

Feature Binding and Robust Vision in Machines and Primates

Niels Leadholm

St. Edmund Hall

University of Oxford

A thesis presented for the degree of

Doctor of Philosophy

Hilary 2022

Abstract

Decades of research have shed light on some of the computational elements that enable the extraordinary visual capabilities of primates, yet the brain's solution to the *binding problem* remains unresolved. In particular, how are the relations of separately encoded features represented in a meaningful way? More recently, machine vision has experienced significant advances with the proliferation of deep-learning techniques, although robust object recognition comparable to primate vision remains out of reach. We propose that these two unresolved goals may be related, leveraging recent theoretical work in computational neuroscience to improve the robustness of machine vision systems. Similarly, we adopt concepts from robust computer vision systems and high-level computational neuroscience models to explore a possible biological implementation of binding at the neural level.

We begin by implementing a network where grid cells are employed as a basis for encoding the spatial relations of features in vision, enabling rapid feature binding for novel sensory information. The utility of this system is then demonstrated on a series of challenging machine vision tasks, including object recognition under saccade-like visual inputs, demonstrating superior performance to traditional machine learning approaches. Further analyses explore the importance of spatial information for the network, and secondary benefits of the architecture such as robustness under continual learning, and the availability of predictive representations. Our results emphasize that this form of binding can serve as a basis for designing enhanced machine vision systems, and may represent computations in the primate brain that account for our current superiority over artificial systems on a variety of visual tasks.

Subsequently, machine vision systems designed for robustness, along with a high-level computational neuroscience model, form the basis for exploring how biological neurons might implement learned feature binding to support the recognition of objects. In particular, we establish the biologically and computationally constrained settings under which a combination of axonal conduction delays, coincidence detection, and Spike-Timing-Dependent Plasticity (STDP) may be sufficient to enable the emergence of object-selective neurons that are both translation invariant and form-specific in their responses. Our combined analysis and simulation results suggest

that such a computation represents a feasible, albeit potentially specialized, operation that the primate brain could leverage for robust object recognition.

Finally, we explore the phenomenon of adversarial examples, suggesting that these may partly be a consequence of the low-dimensional object representations typically assumed in machine vision systems. Leveraging recent developments in theoretical neuroscience, we present a novel deep-learning architecture which approximates the notion of hierarchical binding. Augmented with representations designed to better capture the rich variation of natural objects, the architecture is shown to cause a variety of adversarial attack methods to fail. Additional analyses provide evidence that these results are related to the introduced hierarchical binding, and its influence on the geometry of the network's representations, providing a novel approach for the design of more robust machine vision systems.

Acknowledgements

The research presented in this thesis was funded by the Biotechnology and Biological Sciences Research Council (BBSRC), and I would therefore like to extend my gratitude to both the BBSRC and the public for supporting this scientific inquiry. I would also like to thank the members of the Interdisciplinary Biosciences DTC, including course co-director Esther Becker, for all of their support over the last several years.

Science is, at its greatest moments, a collaborative endeavour. I would like to thank the members of the Oxford Foundation for Theoretical Neuroscience and Artificial Intelligence for their input and support over the last several years, including my supervisor Simon Stringer, as well as Nasir Ahmad, Henry Jordan, Elliot Smith, James Isbister, Daniel Navarro, Toby St Clere Smithe, and Hannah Dury. My time in the lab has been a pleasure, and I will miss the many fascinating conversations we shared. Similarly, I would like to thank my co-supervisor, Mark Buckley, as well as the members of Numenta for the excellent visiting research period I experienced at their company. I would particularly like to thank Subutai Ahmad for his ongoing support and mentoring.

I would like to extend my deepest gratitude to my partner, Jayani Chakravarti, as well as my parents and my sister. This undertaking would not have been possible without all of the love and support I received from them. Similarly, I am indebted to my many friends, including those who are scattered around the world, but with whom I still keep in touch, as well as those that I made during the course of my research, including Felix Homma and Nick Arning. Thank you to everyone for believing in me, and for all the memories we have shared over these last few years.

Table of Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 The Binding Problem and Object Representations . . . 11

1.2.1 Feature Integration Theory . . . 13

1.2.2 Token Creation . . . 15

1.2.2.1 Mechanisms for Representing Tokens . . . 15

1.2.2.2 The Nature of Spatial Information for Token Creation . . . 23

1.2.3 Conjunction Detection . . . 25

1.2.3.1 Convolutional Neural Networks . . . 26

1.2.3.2 The Spatial Sensitivity of Conjunction Detection . . . 27

1.2.4 Hierarchical Binding . . . 28

1.2.4.1 Hierarchical Binding for Novel Relations . . . 29

1.2.4.2 Hierarchical Binding Through Polychrony . . . 30

1.3 Overview of Thesis . . . 36

2 Grid Cell Path Integration For Movement-Based Visual Object Recognition **39**

2.1 Introduction . . . 39

2.1.1 Motivation . . . 40

2.2 Related Literature . . . 43

2.2.1 Grid Cells . . . 43

2.2.2 Cortical Models . . . 45

2.2.3 Grid Cells for Object Recognition : Models . . . 45

2.2.4 Grid Cells for Object Recognition : Biology . . . 46

2.2.5 Perception Through Saccades in Humans . . . 47

2.2.6 Multiple Views in Machine Vision . . . 48

2.2.7 Associative Memories . . . 50

2.2.8 Few-Shot and Continual Learning . . . 50

2.3 Methods . . . 52

2.3.1 Overview . . . 52

2.3.2 Sparse Feature Extraction . . . 53
2.3.3 Sensorimotor Network . . . 56
2.3.3.1 Network Architecture . . . 59
2.3.3.2 Learning in the Sensorimotor Network . . . 63
2.3.4 Classification Method . . . 65
2.3.4.1 Learning . . . 65
2.3.4.2 Inference . . . 66
2.3.5 Recall Method . . . 67
2.3.6 Comparison Classifiers and Decoder Network . . . 68
2.3.7 Hyperparameter Selection . . . 70
2.4 Results : Grid Cell Path Integration as a Basis for Robust Object Recognition 70
2.4.1 Inference Given Arbitrary Sequences, and the Problem of Translation Invariance . . . 73
2.4.2 Rapid Inference With Partial Input Sequences . . . 82
2.4.3 The Use of Self-Movement Information and Form Robustness . . . 82
2.5 Results : Additional Capabilities of GridCellNet . . . 88
2.5.1 Recall of Training Examples . . . 89
2.5.1.1 Recall Despite Noise . . . 89
2.5.2 Predictive Representations . . . 92
2.5.2.1 Image Reconstruction . . . 93
2.5.2.2 Associative Memories . . . 96
2.5.3 Continual Learning . . . 97
2.6 Discussion . . . 99
2.6.1 GridCellNet and Binding . . . 101
2.6.2 Grid Cells for Visual Object Recognition . . . 102
2.6.3 Empowering Agents with Flexibility . . . 103
2.6.4 Translation Invariance . . . 103
2.6.5 Recall . . . 104
2.6.6 Predictive Representations . . . 105
2.6.7 Continual, Few-shot, and Rapid Learning . . . 106
2.6.8 Alternative Architectures for Sequential Inputs . . . 107

2.6.9 The Strengths of Deep Learning . . . 109
2.6.10 Additional Data-Sets . . . 110
2.6.11 Conclusion . . . 110

3 Delay-Lines and Robust Translation Invariance 111
3.1 Introduction . . . 111
3.1.1 Typical Approaches to Translation Invariance . . . 112
3.1.2 An Alternative Approach . . . 117
3.1.3 Overview of Translation Invariance Chapters . . . 122
3.2 Relative Spike Times for Robust Object Recognition . . . 123
3.3 The Emergence of Spatial-Configuration Functions . . . 141
3.3.1 Monotonic Firing Rates . . . 143
3.3.2 Travelling Waves . . . 146
3.4 Preview of Analysis Results and Methods . . . 148
3.5 Methods . . . 150
3.5.1 Network Architecture . . . 151
3.5.2 Differential Equations . . . 153
3.5.2.1 Cell equations . . . 153
3.5.2.2 Synaptic conductance equations . . . 153
3.5.2.3 Synaptic learning equations . . . 154
3.5.2.4 Hyperparameters . . . 156
3.5.3 The Modelling of Input Feature Neurons . . . 156
3.5.3.1 Modelling the Spatial-Configuration Functions . . . 159
3.5.3.2 Presentation of the Input Spikes . . . 164
3.6 Results : Establishing the Parameter Space . . . 165
3.6.1 The Drift Coefficient and the Temporal Offset of Features . . . 167
3.6.2 Network Connectivity . . . 173
3.6.2.1 Dense Networks . . . 173
3.6.2.2 Sparse Networks . . . 179
3.6.3 The Integration Window . . . 187
3.7 Discussion . . . 194

4 Translation Invariance in Simulated Spiking Neural Networks 198

4.1 Introduction . . . 198

4.2 Related Literature . . . 198

4.2.1 Temporal Computations With Delay-Lines in Biology . . . 198

4.2.2 Spiking Neural Networks : Delay-Lines and Object Recognition . 200

4.3 Methods . . . 204

4.3.1 Training and Evaluation . . . 204

4.3.2 Hyperparameters . . . 205

4.3.3 Performance Measures . . . 206

4.3.3.1 Information Theory . . . 206

4.3.3.2 Linear Classifier Accuracy . . . 207

4.4 Results : Within-Distribution Translation Invariance . . . 208

4.4.1 Performance Across Parameter Space . . . 208

4.4.2 Dense Networks as Approximations of Larger, Sparse Networks . 213

4.5 Results : Online Translation Invariance . . . 217

4.5.1 Observing Online Translation Invariance in the Base Setting . . . 218

4.5.2 The Basis of Robust Representations . . . 225

4.5.2.1 The Role of STDP for Robustness . . . 225

4.5.2.2 The Significance of Object Functions and Delay-Lines . 227

4.5.3 The Failure Case : Drift Coefficient of 0 . . . 229

4.5.4 The Significance of the Integration Window . . . 231

4.5.5 Performance Given Higher Amounts of Jitter . . . 236

4.5.6 A More Conservative Drift Coefficient . . . 237

4.5.7 Preserved Delays and the Relation to Experimental Work . . . 240

4.6 Discussion . . . 241

4.6.1 Spatially-Sensitive Conjunction Detection as a Specialized Computation . . . 242

4.6.2 The Significance of Myelin Plasticity . . . 246

4.6.3 Potential Limitations . . . 249

4.6.4 On The Sufficiency of Pooling-Based Conjunction Detection . . . 252

4.6.5 Experimental Predictions . . . 253

4.6.6 Conclusion 254

5 Hierarchical Binding in Convolutional Neural Networks: Making Adversarial Attacks Geometrically Challenging 255

5.1 Introduction 255
5.1.1 Motivation 256
5.2 Related Literature 259
5.2.1 Hierarchical Binding 259
5.2.2 Preserving Low-Level Information in Computer Vision 261
5.2.3 Adversarial Examples 262
5.3 Model Description 263
5.3.1 Approximating Hierarchical Binding 263
5.3.2 Model Architectures 268
5.3.3 Regularization 268
5.3.4 Additional Model Details 270
5.4 Evaluation Methods 273
5.4.1 Adversarial Attacks 273
5.4.1.1 Gradient-Based Attacks 275
5.4.1.2 Decision-Based Attacks 276
5.4.1.3 Transfer Attacks 276
5.4.1.4 Hyperparameters for Adversarial Attacks 278
5.4.2 Geometrical Measures 279
5.4.2.1 Measuring Manifold Geometry 279
5.4.2.2 Eigenspectrum Analysis 282
5.5 Results : Hierarchical Binding Causes Multiple Adversarial Attacks to Fail 283
5.5.1 Hierarchical Binding Alone 283
5.5.2 Hierarchical Binding with Regularization and Noisy Training Data 285
5.5.3 Comparisons to Additional Models on MNIST 290
5.6 Results : Connecting the Geometry of Representations to Hierarchical Binding and Robustness 293
5.6.1 Manifold Extent and Robustness 293

5.6.2 Power Law Scaling of Variance 297
5.6.3 Hierarchical Binding and the Restriction of the Adversarial Space 302
5.6.3.1 The Importance of Binding Dimension and Causality . 302
5.6.3.2 Hierarchical Binding Reduces the Probability of Finding Vulnerable Decision Regions 305
5.7 Results : Performance On Other Data-Sets 307
5.8 Discussion 309
5.8.1 Models That Are Difficult to Fool, vs. Models That Are Impossible to Fool 313
5.8.2 Other Methods That Preserve Low-Level Information 315
5.8.3 Adversarial Examples and Human Perception 316
5.9 Conclusion 319

6 General Discussion **320**
6.1 Interactions Between the Explored Computations 323
6.2 Where is the Solution to Binding Likely to be Found? 324

1 | Introduction

1.1 Motivation

The binding problem of vision describes the mystery of how our brain processes and represents features and their relations in a meaningful way [von der Malsburg, 1981, Hinton et al., 1986, Hummel and Biederman, 1992, Treisman, 1996, 1999, Von Der Malsburg, 1999]. For example, if viewing the two coloured shapes in Figure 1.1i, how does the primate brain appropriately "bind", and thereby represent these as a blue, five-sided shape in the upper half of our visual field, and a yellow square in the lower half [Rosenblatt, 1961, Von Der Malsburg, 1999]? Achieving a correct association of features and spatial information can be used to inform object recognition directly (e.g. "identify if there is a yellow square"), or might inform more complex object recognition based on the spatial relations of a familiar stimulus. Such subsequent object recognition is itself a form of binding [Hummel and Biederman, 1992, Treisman, 1999, Elliffe et al., 2002]. For example, in Figure 1.1ii, how are the features and their spatial particularities used to recognise whether they form a house-like object, as opposed to a lamp-like object? Finally, binding in vision can manifest in a hierarchical form [Treisman, 1996, 1999, Von Der Malsburg, 1999, Grossberg, 2013, Eguchi et al., 2018]. For example, in Figure 1.1iii, how does the primate brain represent that the corner labelled A is part of the square, and the corner labelled B is part of the five-sided shape, while both are part of the lamp-like object?

To understand the first, most classical example of binding described above, we note that it is generally accepted that neurons in the brain can be specialized for certain representations. Whether neurons are highly specific in their response profiles, or can respond to a subset of features, the resulting notion is that at any given time, the identity of the active neurons reflects information about a visual stimulus (where activity might include an increase in the number of spikes a neuron emits every second, i.e. its firing rate). This activity can be conceptualized as a vector of real-valued or binary cells. Thus in the simple example shown in Figure 1.1i, specialised neurons exist for encoding a feature such as the colour blue, or the existence of an object in the upper half of the visual field.

Together, these active cells can unambiguously represent a potentially novel object in a flexible manner. When multiple objects are presented however, this approach becomes problematic. Suddenly, the vector of activity is ambiguous, as this framing does not indicate how the association between two features should be encoded. An intuitive solution is to have an additional neuron that encodes for each possible relation (e.g. a neuron that is active if and only if yellow and square are part of the same object) [Rosenblatt, 1961, Hinton et al., 1986, Von Der Malsburg, 1999]. Indeed, the assumption that all features are encoded by the brain as completely distinct is a significant over-simplification. Many examples exist of neurons encoding multiple "features", such as colour and orientation [Seymour et al., 2009, Garg et al., 2019], complex forms composed of more basic parts [Tanaka et al., 1991, Tanaka, 1996], or even simply a bar together with its position in retinal space [Hubel and Wiesel, 1959, 1962]. However, on the basis that features (of whatever complexity) can at some point exist in a separate form, binding must include a mechanism such that these can be combined and associated in a virtually infinite number of novel ways.

While great strides have been made in understanding how the primate brain processes visual inputs, there remains no established description of feature binding [Humphreys, 2016, Greff et al., 2020]. In addition to our limited knowledge, this partly rests on the fact that, as the examples in Figure 1.1 demonstrate, the binding problem is not a singular problem. As well as a variety of binding operations that must be performed, there are likely multiple ways in which the brain processes and represents any given type of feature relation [Treisman, 1999, Gray, 1999, Hommel and Colzato, 2009, VanRullen, 2009, Humphreys et al., 2009]. Finally, feature binding is not a process limited to vision [Treisman, 1999, Gray, 1999], although this work will be restricted to visual feature binding.

The examples of binding provided in Figure 1.1 can be framed as processing that should both determine an object's identity, as well as capture additional details about the object. This will enable us to define three forms of binding that we will discuss, namely *conjunction detection*, *token creation*, and *hierarchical binding*. Given information arriving on the retina, the brain has to identify the objects that are present. Such an object might represent a statistically regular stimulus, which can be simple (such as a letter "T") or

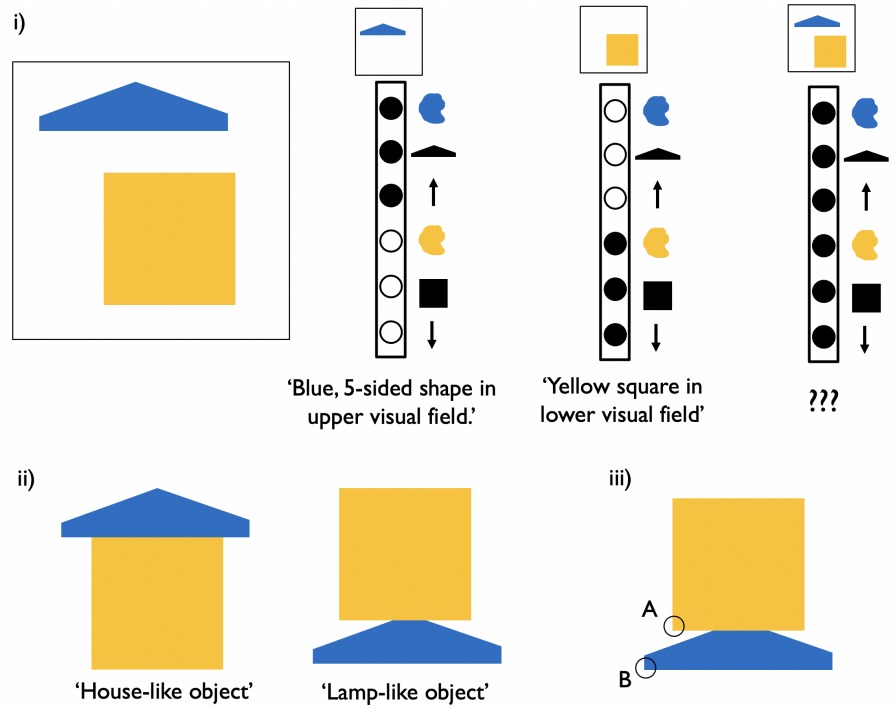


Figure 1.1: *Simple examples of the binding problem in vision.* i) In the presence of multiple stimuli, how does the brain correctly represent associated features (such as the colour yellow with the square shape) in space? This example was originally proposed by Rosenblatt [1961]. Neural signals can encode the presence of particular features, enabling a distributed representation of an object. Such an approach, however, becomes ambiguous in the presence of multiple objects, a phenomenon known as the "superposition catastrophe" [Von Der Malsburg, 1999]. ii) Bound features can form the basis for recognising more complex objects, but this requires an understanding of their spatial relations. Note that even if viewing an apparently singular object such as the house, the form of binding shown in (i) is still not trivial, as there are multiple parts which must each have their properties bound appropriately. iii) The binding problem can also emerge when attempting to capture hierarchical relations. For example, how does the brain represent that corner A is part of the yellow square, and corner B is part of the five-sided shape, but both corners are part of the lamp-like object?

complex (such as a giraffe). In addition, however, the object might be defined by a novel, arbitrary combination of features. Once again, such an object could be relatively simple, such as the coloured shapes in Figure 1.2i, or complex, such as a "purple giraffe with wings" [Treisman, 1998], or a blue bull with pink horns (Figure 1.2ii). We will term the binding used for the identification of a familiar, statistically regular combination of features as *conjunction detection* [Hommel and Colzato, 2009]. Binding that combines at least one, but potentially several, arbitrary features with some underlying spatial representation will be referred to as *token creation* [Treisman, 1999] (sometimes referred to as "dynamic" [Hummel and Biederman, 1992] or "ad hoc" binding [Hommel and Colzato, 2009]). The choice of these terms is motivated by the fact that they can relate to both simple, low-level representations, as well as more complex objects.

Primate vision is typically conceptualized as a hierarchical process, where features of increasing complexity are extracted until the presence or absence of an object-like stimulus, such as a bull (Figure 1.2iii), or the uniqueness of stimuli such as familiar faces, can be determined [DiCarlo and Cox, 2007]. While there is broad agreement on the hierarchical nature of these computations, there is no definitive description of precisely how the lines and colours in the painting in Figure 1.2ii are used by the primate brain to determine that things like a "horn", an "eye", and a "bull" are present. In other words, how is the conjunction detection element of binding (of which object recognition is a special case) actually performed [DiCarlo et al., 2012, Sabour et al., 2017, Hawkins et al., 2019, Tuli et al., 2021]? Furthermore, it is clear that when we look at something like a detailed painting, our brains afford us awareness of far more than just the identity of a particular object. Token creation enables rapidly binding together features with a shared spatial location [Treisman, 1999, Locatello et al., 2020], such as the colour of the bull in Figure 1.2ii.

In addition to such information, however, we have access to an understanding of the hierarchy of visual features that are associated with a given object. This understanding of the relationship between features applies at multiple scales of abstraction. We will refer to this aspect of binding, the third and final one that we will consider, as "hierarchical binding" [Treisman, 1996, Eguchi et al., 2018]. We note that hierarchical binding can apply to relations that are both novel, as well as those that are familiar and statistically

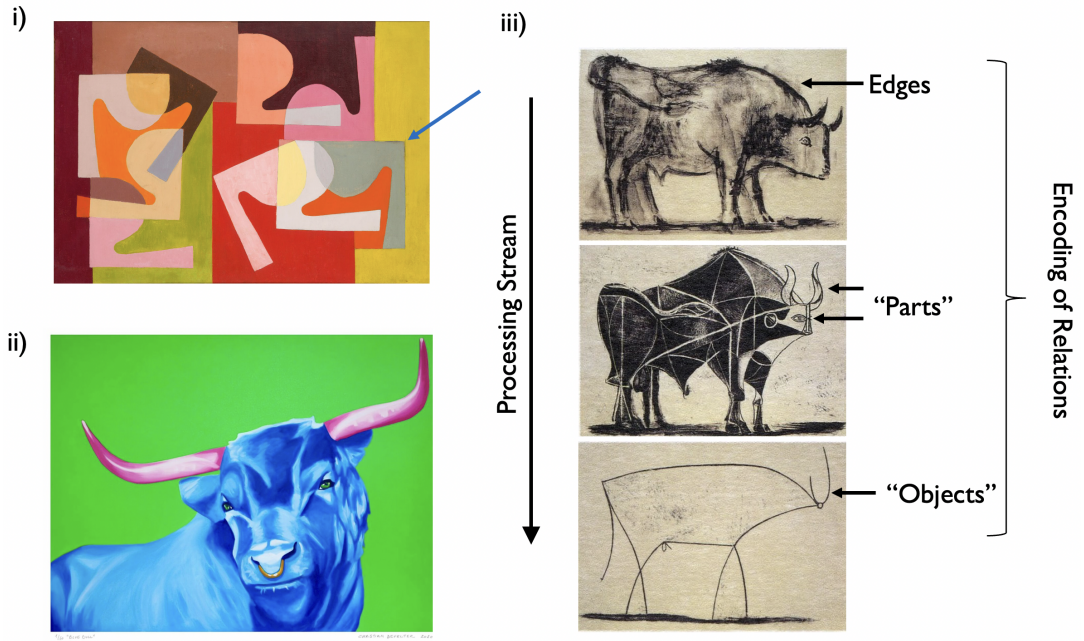


Figure 1.2: *Object recognition and binding.* Shown are a series of reproduced artworks that demonstrate the manifestations of binding in vision. i) An example of binding arbitrary feature combinations into identifiable tokens. This can support, amongst other things, answering a question such as "How many dark brown examples are there of the shape indicated with the arrow?" even as the colours themselves may be somewhat ambiguous ii) Though perhaps somewhat surprising, one can see a blue bull with pink horns for the first time without issue, underscoring the flexibility of binding. iii) A key element of primate vision is the hierarchical extraction of increasingly complex features. This is also typically associated with increased tolerance to distortions of the stimulus (i.e. invariance), such as changes to an object's position on the retina, or its size. Hierarchical binding relates to how the relations amongst these features are captured *and made available to subsequent cognitive processes* (e.g. this lower-level feature is part of the object), rather than just the notion that there is a bull present. The artwork in (i) is *Fractional Module* (oil on canvas), by Saloua Raouda Choucair, 1947-1951; the artwork in (ii) is *Blue Bull* (oil on canvas), by Christian Develter, 2020; the three artworks in (iii) are from *The Bull* (lithographs), by Pablo Picasso, 1945.

regular. Furthermore, the mechanisms that enable hierarchical binding are likely to be closely related to those that can be used in typical token creation and conjunction detection. Finally, it may be that most forms of token creation can be conceptualized as a form of hierarchical binding. For example, the pink in Figure 1.2ii is part of the horns and the bull, but the blue is only part of the animal as a whole. We approach hierarchical binding as a distinct form of binding for two reasons. Firstly, historical theories of token creation typically do not include an explicit discussion of hierarchical relations. Secondly, and perhaps contributing to the first, establishing hierarchical binding of both novel and familiar relations introduces additional challenges, an issue that we will shortly discuss.

With the forms of feature binding that we will consider defined, we now turn to the models that have emerged to describe primate visual processing, and which have thereby explicitly or indirectly sought to provide an account of binding.

Current computational models of vision were built on the seminal work of Hubel and Wiesel on the response profiles of neurons in the visual cortex of cats [Hubel and Wiesel, 1959, 1962], together with experimental work demonstrating a hierarchy of increasingly complex representations in the primate ventral stream and associated regions (V1, V2, V4, TEO, TE, and perirhinal cortex) [Tanaka et al., 1991, Rolls, 1992, Logothetis et al., 1995, Tanaka, 1996, Rolls et al., 1997, Pasupathy and Connor, 1999, Buckley et al., 2001, Brincat and Connor, 2004]. Taking inspiration from such hierarchical processing, successful modelling of primate vision has revolved around the use of local conjunction detectors (often referred to as feature detectors), together with elements that introduce invariance.

Feature detection typically involves a linear transformation of a set of low-level elements, where these elements represent the activity of simpler feature neurons, followed by a non-linear output. The transformation makes use of learned weights specified in the feed-forward connections of a network (i.e. long-term memory), enabling the detection of a feature like a face, given the activity of neurons representing features such as eyes, a mouth, and a nose. Support of online (i.e. to novel locations) translation invariance is then promoted by inducing low-level feature tolerance to translations [Fukushima, 1980, Olshausen and Field, 1996, Wallis and Rolls, 1997, Riesenhuber and Poggio, 1999b, Rao and Ballard, 1999, Elliffe et al., 2002, Stringer et al., 2006, Serre et al., 2007], although

we note that it has become increasingly apparent that such elements are insufficient to fully account for the degree of online translation invariance observed in humans [Bowers et al., 2016, Azulay and Weiss, 2019, Biscione and Bowers, 2021, Blything et al., 2021]. These computational components are repeated in a hierarchical fashion, enabling these systems to recognise the presence of conjunctions (including objects) on the basis of the features that are present and, to varying degrees, their spatial arrangement.

In the machine learning field, architectures that used supervised learning signals adopted similar motifs of hierarchical feature extraction and invariant responses [Lecun et al., 1998, 2015]. With the advent of deep learning, larger architectures could be trained with the back-propagation of error algorithm, resulting in such systems with object-recognition capabilities comparable to humans on a variety of tasks [Krizhevsky et al., 2012]. Such convolutional neural networks (CNNs) have since had an enormous impact on a variety of fields, including enabling the deployment of machine vision systems in challenging, real-world situations such as self-driving vehicles [Bojarski et al., 2016].

Regarding their relation to biological vision, many of the elements on which CNNs rely align well with the known cellular responses that inspired their design, namely the simple and complex cells of Hubel and Wiesel [Hubel and Wiesel, 1959, 1962, Lecun et al., 2015]. Furthermore, these models are capable of predicting neural response profiles in the primate ventral stream [Yamins et al., 2014, Bashivan et al., 2019], as well as some perceptual phenomena in humans [Jacob et al., 2021]. This has led to the adoption of these and related models as leading descriptions of primate vision [Kubilius et al., 2019], and to early arguments that binding can largely be resolved by the conjunction detection employed in these architectures [Riesenhuber and Poggio, 1999a].

Beyond open questions around biological plausibility, such as the use of shared weights and learning mechanisms (although see e.g. Pogodin et al. [2021] and Whittington and Bogacz [2019], Ahmad et al. [2020], Song et al. [2020]), the performance of today's leading systems suggests that they unfortunately remain an incomplete description of primate object recognition and the brain's solution to binding. For example, developing models that efficiently capture a richer representation of objects (i.e. token creation and hierarchical binding), beyond simply the identity of familiar objects, remains an open problem

[Hong et al., 2016, Eguchi et al., 2018, Greff et al., 2019, Burgess et al., 2019, Hawkins et al., 2019, Whittington et al., 2020, Bear et al., 2020, Locatello et al., 2020]. More concerning however, is that current machine vision models remain brittle in a variety of ways, out of keeping with primate vision, and regardless of whether biological plausibility is a consideration or not.

Arguably most well known of these is the vulnerability of these systems to what are known as "adversarial examples", images that have been distorted by targeted noise so as to cause the catastrophic failure of object classification algorithms [Szegedy et al., 2014]. In many instances, the level of noise required to fool a state-of-the-art system is so minimal so as to be imperceptible to a human being. Despite nearly a decade of intensive research on this topic, methods motivated by both biological and engineering principles have had only limited success in preventing adversarial examples [Tramèr et al., 2020b, Croce and Hein, 2020]. A related sub-type of stimuli can be created that are known as "rubbish" (or "fooling") examples, where images that may resemble little more than random noise to a human are confidently classified as an object by state-of-the-art systems [Nguyen et al., 2015, Goodfellow et al., 2015]. As well as being out-of-keeping with human perception, similar stimuli with largely random structure represent the most pronounced gap in the ability of current deep-learning systems to explain neural response profiles in the primate visual cortex [Bashivan et al., 2019].

While adversarial and rubbish examples are well known for their jarring incongruence with human perception, additional challenges persist in the machine vision field. For example, systems remain limited in their ability to meaningfully integrate sequential visual inputs for object recognition, as would be necessary for an agent interacting in a complex environment [Pezzementi et al., 2011, Ranzato, 2014, Seeland and Mäder, 2021]. Furthermore, while CNNs display invariance to novel translations of an object that can be, under appropriate circumstances (i.e. training and design), comparable to human beings [Han et al., 2020, Blything et al., 2021], it remains unclear at what cost such invariance is achieved [Hummel and Biederman, 1992, Olshausen et al., 1993, Sabour et al., 2017, Kamath et al., 2021]. In addition to the sensitivity these architectures display to adversarial examples, their classification performance reveals a non-human bias towards using

low-level features such as textures, rather than object shapes [Baker et al., 2018, Geirhos et al., 2019, Malhotra et al., 2020, Geirhos et al., 2021]. Consistent with this, these architectures can, to varying degrees, be approximated as "bag-of-features" detectors [Brendel and Bethge, 2019]. Finally, these systems generally remain vulnerable to more generic forms of noise and distortion, out of keeping with human robustness [Geirhos et al., 2018, Mu and Gilmer, 2019, Geirhos et al., 2021].

Beyond vulnerability to image characteristics, deep neural networks can also demonstrate brittle learning. For example, achieving strong performance given only a handful of examples ("few-shot learning") remains an open challenge, whereas this is something that humans perform easily [Lake et al., 2015]. "Continual learning", on the other hand, requires a system to learn a novel task while retaining performance on older tasks. The setting of continual learning remains challenging with current neural network models owing to the phenomenon of *catastrophic forgetting*, where previous representations are obliterated to make way for newer ones [McCloskey and Cohen, 1989]. Once again, however, this is a learning setting that humans accommodate naturally (see e.g. Flesch et al. [2018]).

The above challenges can largely be captured by the notion of generalization to instances outside of the training distribution. Under certain settings, engineered systems appear either unable to generalize in the first place, or the mechanisms that do enable generalization contribute to some other subsequent failure.

The current status in the literature therefore converges on a common theme. On the one hand, our understanding of how the brain performs visual feature binding appears incomplete, given the inability of leading models to fully capture primate vision. At the same time, the ability to engineer even biologically implausible systems with comparable robustness to primate vision has not been achieved. The ability of our current artificial systems to capture seemingly many aspects of primate vision suggests that the community has achieved a partial explanation for binding, but that these deficiencies may represent a common gap in knowledge. Thus, novel theoretical work in neuroscience and psychology to explain binding may represent promising approaches for designing machine vision systems with enhanced robustness. Similarly, new algorithms that have been developed

in the machine learning literature to enhance robustness may have inadvertently captured the same computations that emerged in the primate visual system through evolution.

There have recently been renewed calls for ideas from neuroscience and psychology to be used in influencing the design of machine learning systems [Lake et al., 2016, Zador, 2019], consistent with the beneficial transfer of ideas that has occurred in the past [Hassabis et al., 2017]. In the domain of adversarial examples, various efforts to add elements of primate vision such as foveal-like processing [Vuyyuru et al., 2020] and stochastic neuron behaviour [Dapello et al., 2020], as well as to constrain models to more closely match biological activity [Li et al., 2019, Nassar et al., 2020], have made some inroads towards enhancing robustness. Biologically inspired approaches have also been found to enhance robustness to a range of non-targeted perturbations [Evans et al., 2022], and to reduce the use of unstructured information [Malhotra et al., 2020, Tsvetkov et al., 2020]. More broadly, biologically motivated elements have demonstrated utility in settings such as continual learning [Kirkpatrick et al., 2017] and in rapid learning and generalization from sequential sensory inputs [Whittington et al., 2018, Lewis et al., 2019]. This suggests that concepts from biology can provide an avenue towards more robust systems, even if these approaches have yet to close the gap between humans and machines.

At the same time, the efforts of scientists to engineer better machine vision systems can provide theoretical avenues for understanding the computations performed in the primate brain. For example, ResNets are a form of CNN where the addition of skip connections enables the training of deeper architectures, resulting in significant improvements on tasks such as object recognition [He et al., 2016]. Since their creation, ResNets have been useful as a means of conceptualizing the recurrent computations that may be performed in the primate ventral stream [Kar et al., 2019]. As another example of domain transfer, the field of disentangled representation-learning has sought to develop models better capable of generalizing to novel configurations of learned factors [Higgins et al., 2017] (although see Montero et al. [2021]), and indeed recent experimental work has lent support to the existence of such representations in the primate brain [Higgins et al., 2021]. Taken together, it appears worthwhile to explore the insights into primate vision that might come from advances in machine vision.

Motivated by the potentially fruitful interface between neuroscience and machine learning, we set out to explore how recent theoretical work on feature binding [Eguchi et al., 2018, Hawkins et al., 2019, Lewis et al., 2019] could play a role in the design of more robust machine vision systems. Similarly, we explore how recent approaches for developing more robust machine vision systems [Sabour et al., 2017], which share properties with computational neuroscience models of word recognition [Davis, 1999, 2010b], could form a basis for binding in primates by investigating such algorithms at an implementation level. Chapter 2 and Chapter 5 provide evidence of success in the first domain, with Chapter 2 exploring the task of object recognition given sequential visual inputs, and Chapter 5 investigating adversarial robustness. Chapters 3 and 4 will provide evidence that a high-level binding algorithm for online translation invariance could indeed represent a neurally plausible, albeit potentially specialized, element of object recognition in the brain.

1.2 The Binding Problem and Object Representations

In our above motivation, we touched on the nature of the binding problem and its likely importance for understanding the computations that enable robust vision. We now present a more extensive overview of historical theories of binding, as well as recent developments. The robustness challenges that our work explores will be addressed in the chapters in which they arise.

The concept of holistic object perception dates back to the early work of the Gestalt psychologists, who explored the influence on human perception elicited by stimulus properties such as closure and proximity (reviewed in Wagemans et al. [2012]). Subsequently, evidence emerged to suggest that visual processing in the brain was specialized and segregated to separate regions, such as for the identification of colours vs. the orientation of lines [Zeki, 1978]. At a microscopic level, spatially localized neurons in animals such as cats [Hubel and Wiesel, 1959, 1962], and primates [Tanaka et al., 1991, Rolls, 1992, Logothetis et al., 1995] could be identified that increased their activity in response to different features and objects. These findings presented an apparent paradox, where features could be encoded by the activity of distinct neurons or sets of neurons (of which Figure

1.1i shows a simple example), and yet objects are perceived as structured and whole.

At the same time, connectionist models emerged as a means of understanding the computations of the brain [Rosenblatt, 1961, Rumelhart et al., 1986, McClelland et al., 1986]. The binding problem in such models arose owing to the challenge of unambiguously representing the relations of multiple features or objects at the same time, a particularly acute issue in theories arguing for highly distributed representations (parallel distributed processing theory) [Rumelhart et al., 1986, McClelland et al., 1986]. While this assumption could be relaxed in connectionist architectures, it was still necessary to identify a solution that did not result in a combinatorial explosion of conjunction coding units [Hinton et al., 1986, Hummel and Biederman, 1992, Von Der Malsburg, 1999, Treisman, 1999]. In addition to this question of *encoding* relations, uncertainty arose in how to process the spatial relations that exist between separate features in order to robustly determine that a particular conjunction is present [Hummel and Biederman, 1992, Treisman, 1999, Elliffe et al., 2002].

Given the above challenges, much of the early work on binding focused on the ostensibly serial nature of it, where it was argued that ambiguity could be resolved by focusing processing within a narrow attentional window [Treisman and Gelade, 1980]. Since then, evidence has emerged that many neurons can show mixed selectivity, including to classically distinct features such as colour and orientation [Seymour et al., 2009, Garg et al., 2019]. Furthermore, complex feature processing such as object recognition can occur without attention if stimuli are familiar [Li et al., 2002, Fei-Fei et al., 2005]. As such, a more nuanced view of binding has emerged that separates out the processes behind binding for novel conjunctions vs. binding for more familiar objects [Hommel and Colzato, 2009, VanRullen, 2009, Humphreys et al., 2009].

Despite this evolving view, the precise computations and representations involved in binding remain a mystery, even if the binding problem does not necessarily apply to some of the coarser feature relations originally described when the problem was first identified. This development of thinking is captured well by the experimental work that motivated the creation of Feature Integration Theory (FIT) [Treisman and Gelade, 1980], and so we now turn to a discussion of this highly influential work.

1.2.1 Feature Integration Theory

Key to the development of Feature Integration Theory (FIT) [Treisman and Gelade, 1980] was the finding that humans are slow to locate objects defined by a conjunction of simple features, such as red vertical bars amongst horizontal red and vertical green bars. In contrast, stimuli defined by a single, salient feature (such as a red bar amongst green bars) seemingly "pop-out" and are rapidly identified. In the case of the former, there is a significant growth in the time taken to complete a search as the number of distractors increases, while the latter is apparently unaffected. FIT proposed that outside of attention, features were processed in parallel but were relatively free-floating, such that one is incapable of distinguishing objects described by multiple features without first attending to them. Treisman and Gelade [1980] proposed that attention served the role of a spotlight, forming representations in a serial manner where ambiguity was eliminated by only considering features at a given location one at a time.

Alongside additional results from search tasks, such as distinct asymmetries depending on the choice of the distractor vs. target [Treisman and Souther, 1985], FIT was motivated by evidence that humans are susceptible to "illusory conjunctions" [Treisman and Gelade, 1980, Treisman and Paterson, 1984]. In particular, when humans are distracted by a task demanding their attention, they appear to occasionally perceive misbound features, such as the perception of an illusory arrow (←) in the presence of isolated angles (<) and lines (-) [Treisman and Paterson, 1984].

Despite its importance, FIT left a number of questions unanswered. For example, given a complex object composed of multiple parts, Treisman [1999] suggested that each component would be attended in isolation to ensure appropriate binding. It is unclear, however, at what scale a part would be sufficiently localised, raising questions about the time needed to process a complex, multi-part object with nested features. In addition, this approach does not address how the spatial relations between these tokenized parts are processed so as to distinguish objects composed of common features (i.e. the problem of the house vs the lamp shown in Figure 1.1ii). Finally, at a neural level, FIT does not specify the process that attention actually mediates in order to bind the features together [Tsal, 1989, Di Lollo, 2012].

While the above experiments demonstrated the important influence of attention when searching for arbitrary feature conjunctions, additional work increasingly demonstrated that search times could not be easily partitioned into "pop-out" vs slow, serial search times. Instead, they fall on a continuum [Duncan and Humphreys, 1989, Reddy and VanRullen, 2007], and even relatively complex objects defined by the spatial arrangements of their features can be rapidly identified if they are familiar [Wang et al., 1994]. Similarly, evidence grew that object recognition can occur pre-attentively, even for highly complex objects such as animals [Li et al., 2002, Fei-Fei et al., 2005]. Finally, it was argued that some evidence of illusory conjunctions could be explained by experimental design [Donk, 1999]. Even under the assumption that illusory conjunctions are a genuine phenomenon, they appear to be dependent on factors such as the distance between the features outside of attention [Cohen and Ivry, 1989, Ashby et al., 1996], a finding that is difficult to reconcile with the notion of free-floating features before the arrival of attention. As such, while the influence of FIT is difficult to overstate, it is not widely accepted as an accurate theory of brain function [Wolfe, 2020].

The experimental observation that quite complex stimulus processing could occur without attention approximately coincided with the development of massively parallel visual models that could recognise objects in images through the use of learned conjunction detection [Fukushima, 1980, Wallis and Rolls, 1997, Lecun et al., 1998, Riesenhuber and Poggio, 1999b, Stringer et al., 2006, Serre et al., 2007]. Since then, the challenge of developing neural representations of arbitrary feature combinations (token creation) has resurfaced, with current approaches once again relying on either serial processing [Burgess et al., 2019], or slow, iterative algorithms such that features can be appropriately assigned to objects [Greff et al., 2019, Locatello et al., 2020, Bear et al., 2020]. As such, the serial bottle-neck that motivated the original development of FIT appears to remain a relatively accurate portrayal of the constraints that limit token creation, even if this limitation does not apply to all forms of visual feature binding.

With an historical context now provided, we begin by exploring the problem of token creation more generally. This discussion will cover both the methods by which tokens can be represented in neural populations, and the spatial representations that might support

their encoding.

We will subsequently discuss models that are capable of using learning to infer the presence of a familiar feature or object, given the presence of simpler features and their spatial relations (conjunction detection).

Finally, we will discuss approaches to performing hierarchical binding, which has been the focus of interesting work in recent years [Eguchi et al., 2018, Whittington et al., 2018, Hawkins et al., 2019, Whittington et al., 2020, Bear et al., 2020, Hong et al., 2021].

Note that in the following discussion, we have chosen not to separate out the discussion of binding in neuroscience vs. machine learning, as often they are overlapping. Where relevant, however, we highlight the biological plausibility of a given algorithm.

1.2.2 Token Creation

1.2.2.1 Mechanisms for Representing Tokens

Given familiar combinations of features, the success of hierarchical models demonstrates that one can use dedicated neurons or sets of neurons to represent these conjunctions. As seen in Figure 1.1i however, the representation of novel combinations of features presents a unique challenge. Any neurons that are used for conjunctions would need to develop rapidly, and have capacity for re-use, given the combinatorial range of feature combinations that might be realised. At the same time, other mechanisms beyond conjunction neurons might be suitable for representing that a novel combination of features are associated with one another, and distinct from another set of features. How to represent such arbitrary binding at a neural level is central to the question of token creation.

Mechanisms proposed in the literature can broadly be divided into four primary groups. In doing so, we use the terminology from Greff et al. [2020], where these are defined as "instance slots", "sequential slots", "spatial slots", and "category slots". In addition, we deviate slightly from Greff et al. [2020] to argue that binding by synchrony can best be conceptualised as a special form of sequential slots. Owing to the historical and biological significance of synchrony however, we will discuss it separately.

For the models we discuss, we also briefly touch on their approach to *parsing*. In particular, parsing in token creation is the problem of determining which features should actually be associated together as a given token, and segregated from another. Note that this is a separate problem from how to represent the spatial location of a token, as well as how to encode the relations so that they can be read-out by subsequent processing [Treisman, 1999, Gray, 1999]. In general, the majority of the parsing approaches we will discuss rely on a serial or iterative parallel algorithm, in both instances resulting in a slow binding process. This is in keeping with the findings of the experimental literature discussed above, where the binding of novel conjunctions (token creation) appears to be slower than the processing of familiar stimuli (conjunction detection).

Finally, where we discuss machine learning models of token creation, we generally focus on recent approaches that have considered the problem in the setting of unsupervised learning.

Instance Slots The first method for representing tokens is the most straightforward, in which each token has a unique neural representation with a common format (Figure 1.3i). With such an approach, one can ensure that each token is represented independently from others with no risk of interference between them. As such, the representations can be accessed in parallel after they have been formed [Greff et al., 2020]. While the ability to ensure that representations remain distinct is appealing from a computational point of view, it is unclear how well this would explain e.g. the gradual degradation in short-term binding representations that humans experience as a function of object set sizes [Luck and Vogel, 1997].

As separate representations exist for each token, features can also be assigned in parallel with different tokens, however performing parsing in parallel is non-trivial [Greff et al., 2019, 2020]. As such, these methods typically rely on slow, iterative algorithms [Greff et al., 2019, Locatello et al., 2020] such as message passing in graphs [Bear et al., 2020]. Models relying on self-attention, such as transformers [Vaswani et al., 2017] perform parsing in parallel on a single forward pass, but this comes at a significant computational cost, and this is generally not viewed as biologically plausible [Goldstein et al., 2022] (although see also some discussion of the biological plausibility of transformers in Mumford

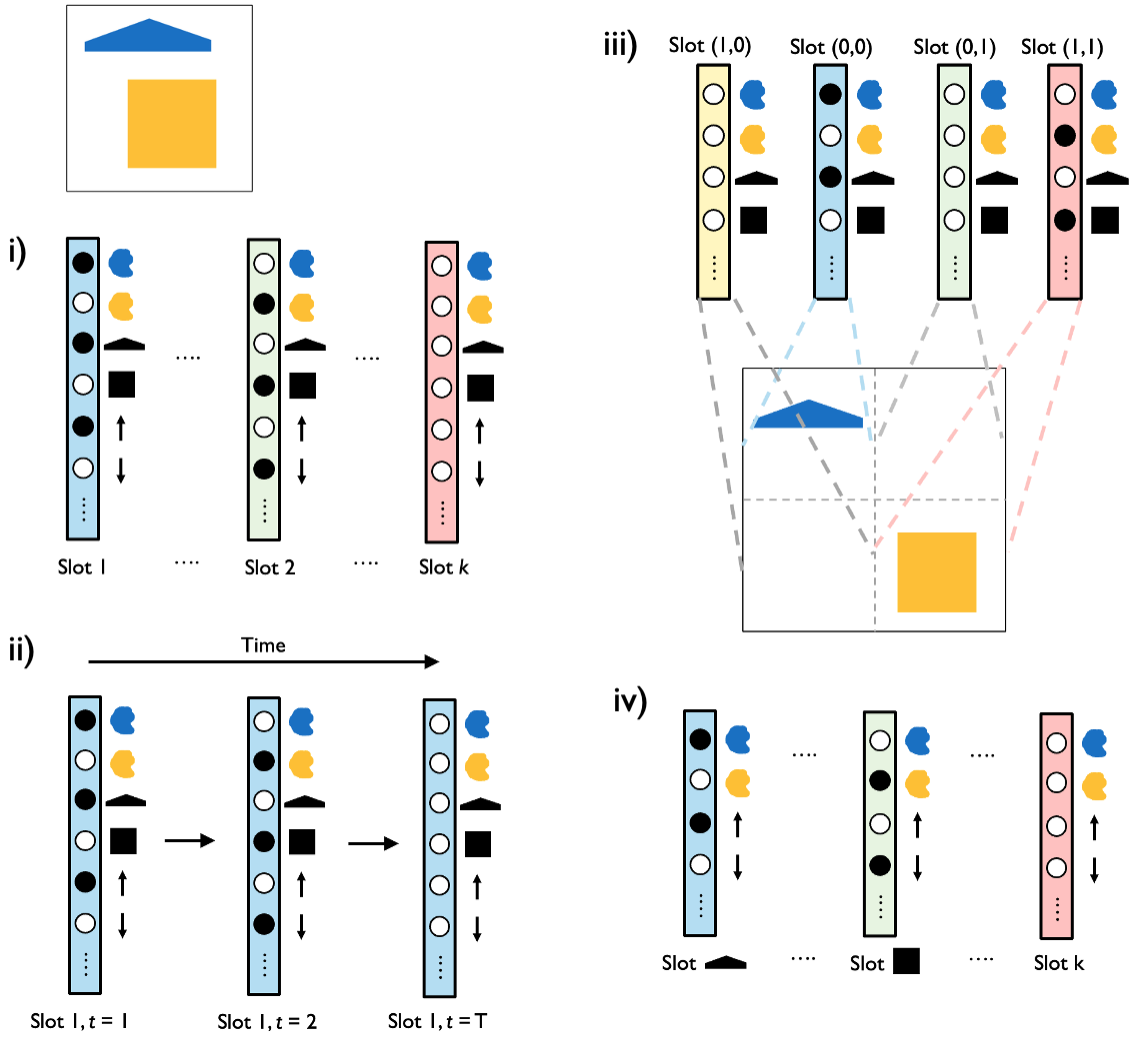


Figure 1.3: *Slot Types for Representing Tokens.* Shown is a diagrammatic representation of the different slot types that can be used to represent multiple arbitrary feature relations (token creation), while avoiding the superposition catastrophe. The simple example stimulus is provided at the inset. The primary approaches are i) instance slots, ii) sequential slots, iii) spatial slots, and iv) category slots. Note that for simplicity, in (iii) we show the slot representation in a setting where the two shapes in the image are clearly confined to separate slots. In reality, there would be numerous receptive fields (i.e. spatial slots) with varying degrees of overlap. We further highlight that binding by synchrony can be viewed as a special case of sequential slots, where time is at the millisecond scale, and representations may repeat (e.g. at a particular phase relative to an oscillatory cycle). Finally, note the implicit encoding of location information in the spatial slots (iii), and of object identity in the category slots (iv). Figure adapted, with permission, from Greff et al. [2020]

[2020]).

Sequential Slots The second approach is to use a common pool of neurons to represent multiple tokens, such as in a recurrent neural network that attends to objects in a sequence [Greff et al., 2020]. In such a situation, the network generally only ever explicitly represents a single token in its activity at any particular time (Figure 1.3ii). As these approaches often rely on a common pool of neurons, elements such as sparsity can ensure that the fidelity with which tokens are preserved is relatively robust [Hawkins et al., 2017, Lewis et al., 2019, Whittington et al., 2020, Hedayati et al., 2022], where the extreme form of sparse representations (a localist code) may be used [Bowers et al., 2014, Manohar et al., 2019]. These approaches have the advantage that they are biologically plausible, and need not specify a priori the number of tokens. As a negative attribute from a computational standpoint, such a method for encoding tokens will degrade with more representations [Hawkins et al., 2017, Manohar et al., 2019], although this might be viewed as a strength given similar limitations in humans [Luck and Vogel, 1997]. Of note, similar approaches have been used in the encoding of *familiar* conjunctions in short-term memory to mitigate the superposition catastrophe [Botvinick and Plaut, 2006, Bowers et al., 2014].

In the above biologically motivated models, parsing is generally handled by limiting the input to a narrow window of space, and assuming that the features that are there belong to a common object or feature-set. This makes parsing of novel feature combinations a straightforward and serial process, akin to Triesman's attentional spotlight [Treisman and Gelade, 1980]. As a disadvantage, however, it is unlikely that such an approach is sufficiently rich to handle the inherent ambiguity of how features can be associated together e.g. in the setting of overlapping objects.

Similar approaches in machine learning include architectures such as Multi-Object Network (MONet) [Burgess et al., 2019], where representations are developed sequentially, such that parts of an input space can be progressively processed by the system. The primary difference between MONet and the approaches discussed above is that there is no requirement for the system to maintain previous representations, and therefore no risk of interference between these. This therefore introduces limits on the downstream use of this information [Greff et al., 2020]. Unlike the above models, MONet makes use of an atten-

tional mask that gradually 'explains away' the input; this is still a serial process, but has the advantage of being able to accommodate more challenging ambiguity in how features should be appropriately parsed, such as when objects overlap.

Binding by Synchrony as a Form of Sequential Slots An approach that can be viewed as a sub-type of sequential slots is to segregate out token association in the phase of temporally precise neural activity. In this setting, the neurons associated with multiple objects are all simultaneously active from a "rate-coded" point of view (i.e. they all have increased activity levels), and so in a typical sequential slot approach, remain unseparated. By the inclusion of precise spike times however, feature neurons that compose a common object can be bound by enforcing their activity to be temporally synchronous. Synchrony can then maintain separation between multiple groups of neurons encoding different objects through the use of different phases of activity [Milner, 1974, von der Malsburg, 1981, Gray, 1999].

An advantage of this approach over standard sequential slots is that it allows, in theory, for the creation of tokens in parallel [Gray, 1999]. Synchrony is also attractive on the basis that it has received some experimental support [Eckhorn et al., 1988, Gray and Singer, 1989], and it has therefore been applied in a variety of models of binding [Hummel and Biederman, 1992, Grossberg, 2013]. One interesting contemporary example was the model of Reichert and Serre [2014], where complex-valued neurons were used to implement an analogous computation to synchrony (i.e. neurons operating in different phases), but applied to deep neural networks. With such an implementation, it was possible to show the utility of synchrony for settings such as overlapping stimuli.

Despite the above arguments for synchrony, one concern at a computational level is that it is not clear how the information that synchrony captures should be read out [Treisman, 1999, Shadlen and Movshon, 1999, Ghose and Maunsell, 1999]. For example, a pair of low-level neurons firing in synchrony with one another might promote activation of a downstream, coincidence detecting neuron, but by invoking additional read-out neurons, much of the benefit that synchrony would provide is lost. Workarounds have been proposed whereby coincidence detecting neurons might respond to only a subset of synchronous inputs, and subsequently resonate with one another, however it remains a

matter of debate how well this avoids the need for dedicated read-out neurons [Shadlen and Movshon, 1999, Von Der Malsburg, 1999]. Thus, while synchrony might still serve a computational purpose, it is not clear the degree to which it resolves the question of token creation [Ghose and Maunsell, 1999]. Furthermore, the introduction of biological elements to models, such as conduction delays, can stymie the emergence of synchronous behaviour, and as such, maintaining synchrony in the face of such biological realities remains a topic of ongoing research [Tikidji-Hamburyan and Canavier, 2020].

At an experimental level, a particularly careful investigation was performed by Dong et al. [2008] to look for evidence of binding by synchrony in awake macaques. In their protocol, synchrony was evaluated between neurons responding to edges that were either part of the same object, or two different objects. Importantly, care was taken to ensure that the receptive fields of the neurons did not overlap, and that the object in question was reasonably complex, rather than being e.g. a single oriented bar. Under such an experimental setting, no significant evidence of synchrony encoding the relations of the low-level features was observed.

Given the above computational issues and mixed experimental support, synchrony has not been embraced as a definitive solution to binding, although it may be that it contributes to binding, or plays a more general supportive role, such as in gating information flow [Reichert and Serre, 2014].

Spatial Slots A third approach to token representation is to use a fixed neural representation, such as a grid in retinal space, where each spatial location has a unique set of neurons dedicated to each token (Figure 1.3iii). This is similar to instance slots, in that multiple objects can be represented in parallel, but it is less flexible in the allocation of objects to space, such as if an object spans multiple spatial lots [Greff et al., 2020].

While CNNs are more typically associated with the conjunction detection form of binding discussed in the next section, they perform token creation using spatial slots in their feature processing. In particular, CNNs contain at each level of processing a grid of multiple feature maps [Lecun et al., 2015]. As such, any particular location in the grid can be associated with a variety of features. In order to handle parsing, CNNs make use of their restricted receptive fields, assigning any features present to a common spatial slot.

While this is a relatively coarse approach to parsing, it has the advantage of being able to operate rapidly and in parallel. Underlining their utility, networks such as relational nets have been able to use CNN token representations to answer image-related questions such as "are there any rubber things that have the same size as the yellow metallic cylinder?" [Santoro et al., 2017].

We have previously noted that CNNs leveraged for object recognition typically include elements to introduce invariance, an example of which is max-pooling (defined later). While such elements help support generalization, they also lead to a coarsening of the spatial code (although interestingly, standard CNNs can still learn to exploit information that enables the localization of objects in the input [Kayhan and van Gemert, 2020]). Furthermore, highly invariant representations would contribute to a re-emergence of the superposition catastrophe, as a single spatial slot would potentially be dedicated to multiple objects. As such, models that use the final feature map of CNNs for their tokenized representations (as opposed to standard object recognition) typically avoid max-pooling [Santoro et al., 2017, Ali Eslami et al., 2018, Locatello et al., 2020].

We further note that spatial slots in CNNs are particularly powerful because any feature learned by the system can be represented at any location, regardless of whether the feature has ever appeared at that location during training. Importantly, however, this capability is a result of weight sharing, a property of CNNs for which a biologically plausible implementation remains an ongoing research question [Pogodin et al., 2021]. This can be distinguished from locally connected networks, where fixed, learned representations exist for different locations [Eguchi et al., 2015]. In such a setting where a low-level feature must generally be learned at every location, the fact that multiple features might become active at a shared location could still form a basis for a spatial slot token. Whether weight sharing or local learning is used, subsequent conjunction detection can make use of this information downstream to perform tasks such as object recognition.

Category Slots The final major method for representing tokens is through category slots [Greff et al., 2020]. In this approach, a given neuron or set of neurons is dedicated to representing a particular feature in a context-independent manner, with some aspect of neural activity carrying additional contextual information (Figure 1.3iv). This separation of the

encoding mechanism used for the identity, presence (i.e. whether it is actually there), and details of a particular feature are key to the concept of category slots. For example, a particular vector of 8 neurons might be dedicated to representing an "eye" feature. The activity of each neuron might then encode some dimension of interest (such as its orientation or position in the receptive field), while the normalized Euclidean length of the vector could encode the probability of the feature actually being present [Sabour et al., 2017]. In a biological setting, the presence of the feature might be conveyed by the binary response or activation level of feature-specific neurons, where the latter would enable additional information about the probability of the feature being present. To encode spatial information, the *time* at which the neuron spikes could then be used [Davis, 2010a,b].

An appealing aspect of this approach is that it may support generalization, as attributes such as the position of a feature can be dynamically associated with dedicated representations for a particular feature identity. Indeed, it can be argued that this supports a form of symbol-like processing, in the sense that a particular feature representation (e.g. the letter "a", or a particular contour) can be re-used in diverse contexts. Such re-use might occur, for example, where the position of a letter within a word varies, e.g. "cAt" vs. "operA", but where the low-level feature's identity does not change [Davis, 2010b, Bowers, 2017]. This is in contrast to spatial slots, where the representation for a feature at one location is separate from the representation at another location. As such, this approach has been adopted in both machine learning [Sabour et al., 2017] and computational neuroscience models of word recognition [Davis, 2010b] to address generalization questions such as translation invariance.

While arguments have been made for how the brain might generate and make use of such representations [Davis, 2010a, Bowers, 2017, Bowers and Davis, 2017], including by members of our lab (J. Isbister, personal communication, July 2018), it remains unclear whether biologically plausible spiking neural networks could actually process such representations to make full use of their potential strengths. In Chapters 3 and 4, we will explore this question, and delve deeper into precisely why such category slots might prove useful for generalization.

1.2.2.2 The Nature of Spatial Information for Token Creation

Above, we discussed the neural mechanisms that might be used to represent associative relationships for arbitrary feature combinations. An additional element of token creation is the spatial basis to which features are bound. In other words, in what spatial reference frame is the association between features (or even the location of a single feature) specified?

In massively parallel CNNs, similar to the retinotopic organization of the primate ventral stream, spatial information is provided by the identity of the neuron that is responding. In other words, the restricted nature of an individual neuron's receptive field enables it to uniquely identify a location in visual space. In a bid to support generalization, however, CNN architectures typically include elements that make feature responses invariant to translations, resulting in a coarser code of space. Such an approach, however, introduces potential complications for maintaining robustness, an issue that we will explore in Chapters 3 and 4.

In machine learning models that encode more flexible positional information, this is typically provided in an external, fixed reference frame. This might take forms such as a 2-dimensional coordinate [Locatello et al., 2020], or a combination of the 3-dimensional coordinate, pitch, and yaw of a camera in a room [Ali Eslami et al., 2018]. One issue with such an approach is the questions it raises about how this spatial information would be directly available to an agent in a natural environment. More importantly, however, the risk of using such coordinates for binding is that representations will not generalize as the external environment shifts.

Given the limitations noted above, what kind of internal reference frame might the primate brain (or any algorithms one would wish to design) use to encode features in order to support robust downstream processing? We previously noted that category slots might be an avenue for flexibly augmenting feature representations with spatial information. In Chapters 3 and 4, we will explore how a spatial code similar to the one explored in [Davis, 2010b] might exist so as to capture the *relative* positions of features in visual space, and the potential strength of such an approach in supporting generalization given

parallel inputs.

While parallel processing of features is invaluable under settings where objects can be visualized in their entirety from a single fixation, the reality is that we (and any real-world agent) view the world through limited sensors. The ability of primates to dynamically interact with their environment via movement of their eyes, head, and body necessitates the ability to represent space via additional means.

One brain region with key roles in spatial processing is the parietal cortex, owing to its role in a variety of tasks such as allocating spatial attention [Corbetta and Shulman, 2002], and relational reasoning about objects [Pohl, 1973, Friedman-Hill et al., 1995, Chafee et al., 2007, Summerfield et al., 2020]. As such, this brain region is often proposed as a primary source of spatial information for token creation [Friedman-Hill et al., 1995, Treisman, 1996, Grossberg, 2013]. That the parietal cortex would play an important role in the spatial information underlying binding is also supported by the existence of Balint's syndrome. In this rare condition, patients with typically bilateral parietal cortex lesions suffer from a constellation of symptoms, including difficulty reporting more than one object in a complex display [Chechlacz and Humphreys, 2014], which can manifest as persistent illusory conjunctions, even when given unlimited time to view a scene. Furthermore, patients with Balint's syndrome can be significantly impaired in describing the relative positions of objects [Friedman-Hill et al., 1995], although intriguingly, it appears that this may be rescued if additional perceptual queues promote processing of the separate 'objects' as a single object [Barton et al., 2007]. These findings suggest that, particularly in the setting of multiple objects, the parietal cortex plays an important role in enabling the brain to process visual space.

Another key brain region for spatial processing is the hippocampal-entorhinal system, a network involved in both spatial navigation and memory [Eichenbaum and Cohen, 2014], and implicated as the basis of "cognitive maps" [Tolman, 1948, O'Keefe and Dostrovsky, 1971]. Important for our current discussion, this brain region has been a focus of interest for its potential roles in enabling rapid conjunction binding [O'Reilly and Rudy, 2001, Eichenbaum, 2012]. Such a role for binding has garnered additional interest since the discovery of grid cells, neurons which become active at multiple, regular intervals of

space [Hafting et al., 2005], potentially providing a spatial scaffold for binding sensory experiences [Whittington et al., 2018, Hawkins et al., 2019, Lewis et al., 2019, Bicanski and Burgess, 2019, Whittington et al., 2020].

While such a form of binding could play a variety of important roles, our interest is in the visual binding problem. It has been suggested that grid cell computations might underlie general sensory processing and object recognition [Hawkins et al., 2019, Lewis et al., 2019], including vision [Bicanski and Burgess, 2019], and experimental studies have found evidence of grid cell-like activity in primates as a function of visual space [Killian et al., 2012, Nau et al., 2018, Julian et al., 2018]. In Chapter 2, we implement a network that performs binding with sequential slots and grid cells as a basis for spatial encoding. In doing so, we will assess the utility of such an approach to token creation in supporting visual object recognition under a few-shot, continual learning setting, where the visual input is a challenging sequence of local image patches. We defer a more extensive discussion of how grid cells actually represent space, and in particular their possible significance for encoding visual space in primates, to the introduction of Chapter 2.

1.2.3 Conjunction Detection

In contrast to the rapid, online form of binding discussed above, it is clear that many binding relations that are encountered will re-occur. This opens up the possibility to learn about these statistically regular relations, and use neural hardware to process and represent them explicitly [Riesenhuber and Poggio, 1999a]. We will return to the concept of conjunction detection in Chapters 3 and 4, but in the following section, we briefly touch on a number of concepts that were not discussed in our opening motivation.

Whereas token creation methods typically rely on a serial bottle-neck to eliminate ambiguity about how novel associations of features should be bound with one another, conjunction detection can use mechanisms such as learned weights to determine whether features in a restricted region of an image represent a particular conjunction. Importantly, such conjunction detection can operate in parallel.

We note that conjunction detection need not assume that a particular dedicated conjunction neuron exists for every familiar stimulus. Instead, neurons may simply preferentially

respond to particular arrangements of features, and thereby in a semi-distributed manner support classification of objects as the end goal [Rolls and Milward, 2000, DiCarlo and Cox, 2007, Cohen et al., 2020]. Naturally, however, the less selective a representation is, the greater the risk of the superposition catastrophe re-emerging [Bowers et al., 2014], and neural responses in the brain are often highly selective [Logothetis et al., 1995, Pasupathy and Connor, 1999]. We further note that computer vision systems often develop quite selective responses in their hidden units [Nguyen et al., 2016].

1.2.3.1 Convolutional Neural Networks

Chapters 2 and 5 will explicitly make use of CNNs, while in Chapters 3 and 4, we will contrast our biologically motivated network to these architectures. As such, it is worth briefly describing the fundamental operating principles of these networks.

A typical CNN architecture consists of convolution operations and a down-sampling operation such as max-pooling. Convolutions apply a linear transformation at local regions across space, where this weighted sum serves as a feature detector, such as for the presence of an edge or an ear. This is subsequently transformed by a non-linearity. If the activity within this local region and the learned weights match well, then a large activation will be output, suggesting the presence of that feature. Using the assumption that features in one part of visual space may also appear in another, the same convolutions are applied repeatedly across space, making efficient re-use of parameters in what is known as weight sharing. Several different convolutions can be applied at any given level of the network in order to detect multiple different features.

Note that this process is an example of the possible interplay of token creation and conjunction detection. In the case of CNNs, token creation assigns features to a particular spatial slot (say at the first layer of processing, where the feature maps could be the Red-Green-Blue colour channels). The token representations are then processed by learned weights to identify conjunctions that are present. The active (i.e. present) conjunctions then form the basis of spatial slot tokens in the next layer.

Max-pooling is used with the aim of providing the model with translation invariance, by taking the largest activation within a given receptive field, and up-projecting that activa-

tion value alone. When applied to the output of feature detectors, this can be thought of as evaluating, for example, if an ear was present anywhere in a given region of the image, without any concern for its precise location. Subsequent feature detectors can take advantage of this invariance by applying a convolution to the outputs of max-pooled values; this convolution (say for detecting the presence of a cat), will be less sensitive to the spatial particularities of the features of the cat, and should in principle be more capable of generalization [Lecun et al., 2015]. We note however that recent evidence has demonstrated that max-pooling and related architectural modifications are generally insufficient for enabling online translation invariance beyond small shifts, with the prior learning of general representations across space playing an important role [Blything et al., 2021, Biscione and Bowers, 2021].

1.2.3.2 The Spatial Sensitivity of Conjunction Detection

As noted above, the use of the max-pooling elements in CNNs should contribute to generalization to novel translations, but the loss of spatial information results in a coarser spatial code. This may in turn contribute to the conjunction detection process being *inappropriately* insensitive to the spatial arrangement of features [Brendel and Bethge, 2019]. What might be an alternative approach?

In Chapter 2, we will examine models that sequentially process visual inputs, using grid cell computations as a scaffold for token creation. Importantly, this will enable us to explore the significance of grid cell *path integration*, that is using self-movement information to update spatial representations, for object recognition that is sensitive to the precise arrangement of features. In principle, the inference algorithm implemented in this network will also serve as a stepping stone towards sequential classifiers that are translation invariant, a concept that we will explore in the chapter.

In Chapters 3 and 4, we explore the significance of neurons that encode spatial information about features in their relative spike times. By doing so, we will examine whether similar approaches employed in capsule networks [Sabour et al., 2017] and higher-level computational neuroscience models [Davis, 2010b] could feasibly be implemented in biologically plausible, spiking neural networks. In this setting, axonal conduction delays

will serve to transform spike times, providing the scope for object recognition that is both spatially sensitive, and invariant to translations.

Both of the above approaches will therefore examine algorithms that might enable the primate brain and machine vision systems to support conjunction detection that is translation invariant, without relying on coarse spatial information as in typical CNNs. In Chapter 2, the mechanism explored will be relevant for processing largely novel (i.e. few-shot learning) objects, where learning and recognition follow a serial, sequential process. By contrast, the algorithm explored in Chapters 3 and 4 will be relevant for processing learned stimuli in a potentially parallel manner, albeit in the context of novel translations.

1.2.4 Hierarchical Binding

In the previous sections, we have discussed token creation, where arbitrary features are bound to a common spatial reference frame, and conjunction detection, where the spatial relations of features are used to identify statistically regular feature combinations. In hierarchical binding, the task is to capture how features *across* different levels of abstraction relate to one another, and in particular the causal chain of the low-level features that have contributed to higher-level features or objects. As posed by Treisman [1999], what is the mechanism for "storing the sequence of synapses that activated a cell, and where its afferent connections came from"?

In both of the previous forms of binding that we considered, and in hierarchical binding, it is likely that the brain makes use of differing methods for handling arbitrary feature relations vs. statistically regular feature relations. As before, the former is likely to be slower, but efficient in its use of neural hardware, and flexible in its application. The latter, on the other hand, could potentially operate in parallel, albeit while being dependent on learning, and at a significant expense in the quantity of neurons that must be dedicated. As we will now explore, however, unique challenges emerge in the context of hierarchical binding.

1.2.4.1 Hierarchical Binding for Novel Relations

When attempting to capture novel relations in hierarchical binding, an immediate challenge emerges from the fact that multiple subset-superset relations can exist. Unlike in standard token creation, where the desire is to segregate out discrete objects from one another, the divisions within hierarchical binding are less clear.

To better demonstrate this issue, we return briefly to the concept of using synchrony for binding. While binding-by-synchrony could be well suited to grouping together coarse sets of features (e.g. two particular eyes and a nose belong to one face, and not another), it is not clear how this can encode the richer relations that emerge in hierarchical binding. For example, how can a single clustering mechanism be used to represent that a contour is part of a nose, and that the nose is part of a face, while also keeping the attributes associated with the low-level features distinct from one another? Similarly, how can synchronous groupings represent that a nose is part of a face, but not an arm, even though both are part of a body? While it is possible to simply bind all low-level features together with a high-level object [Grossberg, 2013, Hedayati et al., 2022], such an approach does not capture this richer hierarchical structure.

This limitation is not only out of keeping with the complex hierarchy of feature relations that we can conceptualize, but also leads to ambiguity about the computational roles for which synchrony is invoked. For example, in Hummel and Biederman [1992], synchrony of differing phases is used to bind together (and thereby separate) the attributes corresponding to the different *parts* of an object. In contrast, in Grossberg [2013], a single synchronous assembly is proposed to bind together all of the parts of an object together with the object itself. How might this role for synchrony change for different objects, where the notion of "object" might be as diverse as a nose, a head or a "man on a horseback" [Marr, 1982]?

Owing to this challenge, we will observe that some, but not all, of the methods discussed under token creation might be useful for hierarchical binding for novel relations. We note again that most token creation might in fact be at some level hierarchical [Brady et al., 2011], which could align with recent evidence that processes such as working memory

are influenced by hierarchical structure [Nie et al., 2017]. However, owing to the fact that most approaches to token creation do not consider an explicit hierarchy, as well as the unique challenges of hierarchical binding, we have elected to discuss token creation and hierarchical binding for novel relations separately.

Despite the challenge of arbitrary hierarchical binding, recent work has fortunately provided promising methods by which hierarchical structures could be represented, encoding novel feature relations across multiple scales of detail. These have included the use of hierarchical graphs [Bear et al., 2020], and grid cell maps of varying scales to encode feature locations [Whittington et al., 2018, Hawkins et al., 2019, Whittington et al., 2020]. Importantly, these approaches provide scope for querying the relations between features at different levels of abstraction. While these are promising ideas, they are currently limited to a conceptual stage [Hawkins et al., 2019], synthetic data [Whittington et al., 2018, 2020], or require dozens of iterative steps to reach convergence [Bear et al., 2020]. It also remains unclear how exactly the brain would undertake the graph construction process performed in Bear et al. [2020]. As such, understanding how the brain or artificial agents can efficiently represent arbitrary hierarchical binding relations remains an open problem, and one that lies beyond the scope of the present work.

1.2.4.2 Hierarchical Binding Through Polychrony

The impressive capabilities of hierarchical models for object recognition have demonstrated the power of learned conjunction detection. It is thus natural to question whether similar representations could emerge for hierarchical binding relations, and indeed such a possibility was considered some time ago [Von Der Malsburg, 1999]. It later became evident however, that models in which neurons signal information through their firing rates would likely be insufficient for hierarchical binding [Eguchi and Stringer, 2016]. To enable hierarchical binding that could be performed in parallel with learned relations, Eguchi et al. [2018] argued that a combination of precise spike-timing together with hierarchical and recurrent processing would be necessary. A discussion of hierarchical binding through polychrony will introduce a number of concepts that will be useful for Chapters 3, 4, and 5, and so we dedicate a more extensive overview to it here.

As noted under binding-by-synchrony, while some theories of perception argue that only the firing rate of a neuron carries meaningful information (rate-coding), consideration can also be given to the precise times at which neurons spike, resulting in a temporal code. Furthermore, when neurons spike, the discrete action potentials they generate must travel down the cell's axon to reach other neurons. In many theoretical models, axonal conduction delays are avoided or made uniform [Diesmann et al., 1999]. Importantly, however, experimental evidence indicates that both local and non-local axons can be associated with significant conduction delays [Swadlow, 1994, Egger et al., 2020]. Finally, the spikes that arrive on an afferent neuron exert influence via the strength of a synaptic connection (its "weight"). In biological neurons, the strength of this influence is modified over the course of learning via Spike-Timing-Dependent Plasticity (STDP) [Markram et al., 1997, Song et al., 2000]. In STDP, a weight is updated dependent upon the precise difference in time between when the signal arrives and when the output neuron spikes. In particular, if the presynaptic spike arrives shortly before the postsynaptic spike, it implies a causal relation, and the strength of the weight is increased. If, on the other hand, the presynaptic spike arrives shortly after a postsynaptic spike, it implies the presynaptic neuron was merely co-active by chance, and the weight is decreased.

Introducing delays into spiking neural network models, along with modification of weights through STDP, can cause neurons to form the temporal code known as *polychrony*. In contrast to synchrony, polychrony is defined by neurons spiking with asynchronous, albeit fixed, temporal offsets [Izhikevich, 2006]. Given appropriate axonal conduction delays, a neuron sensitive to coincidence detection can respond uniquely to a particular polychronous pattern of incoming spikes [Hopfield, 1995]. If the spike-timing of a neuron carries additional spatial information (e.g. the relative location of a feature in a receptive field), then this can enable object recognition that is sensitive to the configural arrangement of features. In Figure 1.4 a simple example of such a polychronous network is demonstrated, although this concept and the biological plausibility of such circuits will be explored in greater detail in Chapters 3 and 4.

We note that polychrony is closely related to the concept of synfire braids owing to the presence of mixed delays [Bienenstock, 1995]. In synfire braids, however, it is as-

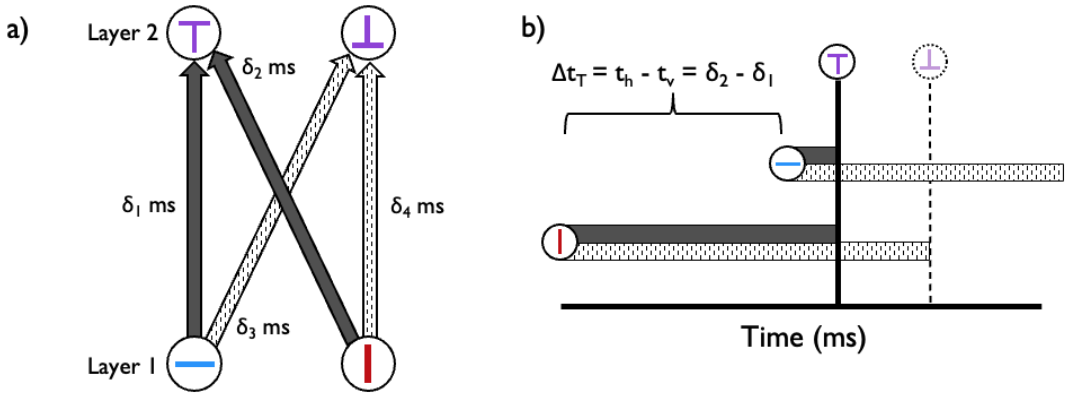


Figure 1.4: *A Simple Polychronous Network.* a) In this network, low-level features send spikes to higher-level features via connections that are associated with non-uniform conduction delays (δ_i). b) A particular spatiotemporal pattern is described by the differences in spike times of the low-level neurons ($\Delta t_T = t_h - t_v$), where t_h and t_v are the spike times of the horizontal and vertical bar neurons respectively. If this spatiotemporal pattern offsets the associated delays (i.e. $\Delta t_T = \delta_2 - \delta_1$, or equivalently, $t_h + \delta_1 = t_v + \delta_2$), the spikes will arrive nearly simultaneously at a particular output neuron, enabling specific encoding of a particular conjunction. Note that we assume here that the spike-timing is a function of the positions of the features, and as such, the signals in this case do not arrive coincidentally at the $\perp$ neuron. We further note that in this case we are not concerned with rotation/reflection invariance, and thus treat T and $\perp$ as different objects, as is the case in the Roman alphabet for the letters b/d/p/q.

sumed that neurons will eventually result in a volley of spikes that are all simultaneous (i.e. synchronous), and that this synchronous wave alone carries information. In polychrony, no such synchronous patterns are required to emerge, and rather the asynchronous, temporally-fixed patterns are themselves the basis of information in the network.

We now return to the question of hierarchical binding. In a typical rate-coded neural network, a translation invariant neuron representing a higher level feature (perhaps a "T") increases its firing rate given the presentation of the stimulus, regardless of its position on the retina (see e.g. the rate-based model in Elliffe et al. [2002]). The high-level feature neuron does not, however, capture the identity and spatial information of the low-level neurons that drove its representation, and therefore its precise low-level constituent features. Figure 1.5 shows a circuit where a high-level neuron has learned to encode a T at multiple locations. In this setting, the standard conjunction detection can convey that a T is present, but there would be a loss of information about the particular low-level features that composed that T, as opposed to, say, the low-level features forming a $\perp$ at the

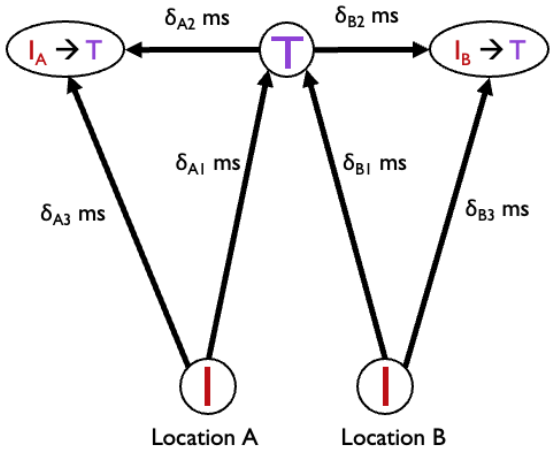


Figure 1.5: *Hierarchical Binding Through Polychrony.* The T neuron in this setting has learned to respond to the object at either of two different locations, A and B, providing invariance (note that for simplicity, the horizontal bar neurons are not shown). When the vertical bar neuron at location A drives the T neuron to fire, spikes from the T neuron and $|_A$ neuron will arrive simultaneously at the associated binding neuron if the relationship $\delta_{A3} = \delta_{A2} + \delta_{A1}$ approximately holds. Importantly, if another object is simultaneously presented at Location B (say a $\perp$), the vertical bar neuron at Location B can be co-active. However, the associated binding neuron will not be activated, as the signals arriving at the T neuron from $|_B$ will not have subsequently triggered the T neuron's spike. Note that if the activity of the neurons is fully described by firing rates, with no consideration for the coincidence detecting properties of neurons, then no such guarantee exists. Instead, elevated signals will arrive at both binding neurons in the circuit, resulting in a form of superposition catastrophe in which hierarchical binding information is lost. Finally, we note that in real biological circuits, each neuron is likely to be represented by a polychronous neuronal group, rather than an individual neuron, and the connections shown may be multi-synaptic.

other location. In the spiking network proposed in Eguchi et al. [2018] however, a key relationship was proposed. Given the correct conduction delays, together with top-down and lateral connectivity, coincident signals can arrive at a *binding neuron* from the low-level and high-level feature neurons. Importantly, the arrival of signals will temporally coincide if, and only if, the low-level feature actually caused the higher level neuron to spike. In this way, the information about what low-level features compose more complex representations could be captured explicitly in the identity of active neurons.

Note that coincidence detection and the conduction delays serve an important role, as simply having a hard-coded conjunction neuron is insufficient. In particular, given a network using only rate-coded information, the simultaneous presentation of both a T and $\perp$ at the two locations will result in the co-activation of all conjunction neurons. Finally,

STDP can serve the role of pruning synapses associated with irrelevant delays, enabling the necessary delay configurations (such as those shown in Figure 1.5) to emerge [Eguchi et al., 2018]. In particular, as STDP is sensitive to the temporal alignment of signals, it can strengthen weights associated with coincident inputs, while weakening those associated with unrelated (and temporally dispersed) spikes. We note, in addition, that it may be that the modification of the myelin sheaths of axons (i.e. myelin plasticity) could serve an additional role by changing delay times directly [Bowers, 2017, de Faria et al., 2021].

A strength of this proposal is that it provides an explicit, neural basis for encoding relations among features, which makes more straightforward any questions regarding readout by higher levels of processing. This encoding is also capable of capturing multiple, potentially nested levels of relations, unlike a mechanism such as synchrony. The projection of such information to higher layers of the visual hierarchy was dubbed the *holographic principle* [Eguchi et al., 2018]. This upward projection of information would be consistent with the availability of hierarchical and low-level information to higher cognitive functions [Kim and Feldman, 2009, Kang and Maunsell, 2020]. That this process can happen in parallel might help explain why it is apparently more efficient to look for colour conjunctions when they follow a hierarchical relationship [Wolfe et al., 1994]. Finally, the hierarchical binding representations would, owing to lateral and top-down delays, *follow* rather than precede, awareness of the high-level representations of an object, consistent with many experimental results of human perception (see e.g. Hochstein and Ahissar [2002]).

An example of a possible binding neuron that has been identified in neurophysiological experiments is border-ownership cells, which have been identified in the V1, V2, and V4 regions of the primate visual cortex. These cells have a small classical receptive field, similar to bar/edge detecting 'simple cells'. Unlike a classical simple cell however, their response is modulated by what side of an object they form an edge of. In particular, border-ownership cells encode when a particular low-level feature is part of a specific side of a high-level object, rather than simply that a low-level feature is present [Zhou et al., 2000].

Despite the above appealing qualities and the highlighted experimental evidence, it may

be that the number of such possible relations is too numerous to be represented or have computational relevance [Von Der Malsburg, 1999]. Similar concerns were associated with the theories of conjunction representations for feed-forward computations when these were initially proposed [Hinton et al., 1986, Hummel and Biederman, 1992, Treisman, 1999, Von Der Malsburg, 1999]. A semi-distributed code (polychronous neuronal groups or PNGs) would have a much higher representational capacity, providing a route for a sufficiently expressive substrate [Izhikevich, 2006]. Additional constraints, such as the majority of these representations being found in privileged regions of the visual field [Eguchi et al., 2018], or for particularly familiar stimuli, could further limit this potential issue. Finally, the need for a large neural population to support such representations might partially explain the volume of the primate brain dedicated to visual representations, including in particular neurons with smaller receptive fields [DiCarlo et al., 2012].

An additional concern is that the degree to which the brain actually depends on temporal computations remains a matter of debate, with arguments frequently advanced both for [Luczak et al., 2015, Egger et al., 2020, Williams et al., 2020], and against [Baker and Lemon, 2000, London et al., 2010] this possibility. Consistent with this ongoing debate, clear experimental evidence of polychronous neuronal groups in the visual cortex does not currently exist. Such evidence is challenging to acquire, as despite significant advances in the ability to record the activity of neurons in parallel [Stevenson and Kording, 2011], current methods remain limited in comparison to the scale of the visual cortex.

More fundamentally, methods to detect the existence of polychronous patterns are computationally demanding, and suffer from limitations such as in the inability to handle instances of single neurons participating in multiple PNGs. Recently proposed techniques [Russo and Durstewitz, 2017, Quaglio et al., 2017, Van Der Meij and Voytek, 2018] can take days to search for small PNGs in a recording set of only a few dozen neurons, and they suffer from significant scaling issues. These remain challenging obstacles to overcome given current computing and recording technology. As such, the proposal in Eguchi et al. [2018] remains an interesting mechanism by which hierarchical binding could operate in parallel in the brain, at least for familiar hierarchical relations. Whether such an operation actually exists in the brain remains to be seen.

In Chapter 5, we will use the proposal that the brain might use such explicit neural representations of hierarchical relations to explore a novel approach to the challenge of adversarial examples. In particular, we will examine the connection between hierarchical binding, the dimensionality of representations, and the geometry of the representational space that adversarial examples exploit. We will also look for indirect evidence that hierarchical binding might be carried out in parallel by binding neurons by determining whether networks that coarsely approximate these computations better match primate capabilities in vision.

In Chapters 3 and 4, while hierarchical binding will not feature, the non-uniform axonal conduction delays seen in biology, STDP learning, and the coincidence detection properties of neurons will all play a significant role.

1.3 Overview of Thesis

We now briefly summarise the primary contributions of our thesis, focusing on the relationship of each chapter to the concepts introduced above. A more thorough break-down is provided in each chapter:

Chapter 2 - Grid Cell Path Integration For Movement-Based Visual Object Recognition

- We implement a network that performs token creation given saccade-like visual inputs, making use of a sequential slot encoding mechanism, where spatial information is encoded in grid cell activity.
- This network is used to explore the significance of grid cell path integration for enabling object recognition that is sensitive to the spatial arrangement of features, in spite of the sequential nature of visual inputs (serial conjunction detection).
- We also explore the architecture's capabilities for leveraging predictive representations, recalling training examples, and maintaining robustness under few-shot and continual learning settings.
- A subset of this work was published at the British Machine Vision Conference

2021 [Leadholm et al., 2021], where it was selected for an oral presentation (<5% of accepted papers).

Chapter 3 - Delay-Lines and Robust Translation Invariance

- We explore the properties of spiking neural networks where features are represented via category slot tokens that encode feature identity alongside spatial information in their spike times.
- We describe a general framework for how the relative spike times of neurons might enable object representations that are both translation invariant to novel locations, while also remaining sensitive to the spatial arrangement of features (parallel conjunction detection).
- Using a combination of biological and computational constraints, we leverage a variety of analyses to estimate the bounded hyperparameter space in which such encoding could take place.

Chapter 4 - Translation Invariance in Simulated Spiking Neural Networks

- Building on the analysis in the previous chapter, we carry out simulations to evaluate the settings under which STDP learning enables translation invariance to locations experienced during training, and determine under what conditions this fails.
- Using the above results, we estimate an appropriate parameter region within which to evaluate the ability of a spiking neural network to, following STDP-based learning, solve the strictly harder task of online translation invariance, whilst retaining form-specific response profiles.
- Our combined simulations demonstrate that STDP in spiking neural networks may indeed be sufficient to observe the proposed computations, but that this occurs in a narrow parameter range which, although biologically plausible, may limit the extent to which the computations could exist in the primate brain given STDP as a sole learning mechanism.

Chapter 5 - Hierarchical Binding in Convolutional Neural Networks: Making Adversarial Attacks Geometrically Challenging

- We implement a convolutional neural network that coarsely approximates the representations of hierarchical binding by polychrony, capturing the causal relations between low-level and high-level features.
- We demonstrate that this novel architecture introduces robustness under a variety of adversarial attack conditions, in particular to challenging black-box (gradient-free) attack methods.
- Using several analyses, we demonstrate the relationship between hierarchical binding, the geometry of a network's representations, and enhanced robustness to adversarial examples. In doing so, we eliminate a series of alternative causes for the observed effects.

2 | Grid Cell Path Integration For Movement-Based Visual Object Recognition

2.1 Introduction

In this chapter, we will consider the challenging task of recognising objects given a series of local glimpses across visual space. In particular, we will consider the possible importance of grid cells in both enabling such recognition, as well as in supporting other computations such as predicting unsensed regions of the input. Grid cells enable the brain to model the physical space of the world and process transitions in an environment via path integration, updating self-position using information from self-movement. Recent proposals suggest that the brain might use these mechanisms to understand the structure of objects in diverse sensory modalities, including vision. In machine vision, object recognition given a sequence of sensory samples of an image, such as saccades, is a challenging problem when the sequence does not follow a consistent, fixed pattern - yet this is something humans do naturally. In this chapter, we explore how grid cell-based path integration in a cortical network can support reliable recognition of objects given an arbitrary sequence of inputs. Our network (GridCellNet) uses grid cell computations to integrate visual information and make predictions based on movements. We use local Hebbian plasticity rules to learn rapidly from a handful of examples (few-shot learning), and consider the task of recognizing MNIST digits given a sequence of image feature patches. Extending beyond the current literature, we show that GridCellNet can reliably perform classification, generalizing to both unseen examples and completely novel sequence trajectories. Furthermore, by utilizing grid cells for an internal reference frame derived from sensory inputs and internal motor information alone, the classification process represents a significant step towards enabling translation invariance in sequential classifiers. Importantly, classification is dependent on processing the spatial relations between features, conferring robustness to meaningless feature arrangements, consistent with human perception. In addition to its classification capabilities, we demonstrate that GridCellNet is able to predict unsensed regions of the input, that inference can be successful after sam-

pling a fraction of the input space, and that a natural benefit of the proposed architecture is robustness in the context of continual learning. Together, these results demonstrate the value of grid cell computations as an inductive bias in machine learning models. Due to the general consistency of GridCellNet with the known neuroanatomy of the primate brain, it may also serve as a useful model of biological vision across multiple saccades.

Attribution: The following work, and in particular all results relating to the "Rich Local Feature" setting (introduced under Methods), was published in Leadholm et al. [2021]. All of the included material was written and produced by myself, with supervisory roles from co-authors Marcus Lewis and Subutai Ahmad, and from whom I have received permission to include this material in my thesis.

2.1.1 Motivation

When exploring a visual scene, primates sample the world in a serial sequence by performing rapid eye movements known as saccades [Yarbus, 1967]. For the purpose of recognising objects, it is non-trivial that this sampling can follow an arbitrary sequence order, and begin on any part of the object. For example, one might selectively attend to the most salient parts of a face rather than performing a raster scan across the image. While many previous efforts to model primate object recognition have focused on massively parallel processing of a single input, the challenge of dealing with the necessarily sequential nature of sensory inputs has received less attention [Bicanski and Burgess, 2019]. Recurrent-neural networks can perform complex tasks with sequential inputs, and might seem like a natural candidate for such a challenge, yet they struggle to learn when provided with sequences that do not follow a fixed order during training and inference. In the natural world, there are additional challenges that can present themselves. Often only a handful of object examples are available, learning should be rapid (i.e. requiring limited training on the few examples given), and representations should be robust in the face of future learning. These are all constraints that humans are able to accommodate. Understanding how learning and inference under these conditions might be achieved has two appealing aspects. As well as potentially uncovering the basis for human performance in this domain, the flexibility to operate under such a regime could also enable artificial

agents to explore the world in a more principled and adaptive manner.

While recurrent neural networks do not have explicit mechanisms for dealing with this challenge, grid cells might provide a neurally plausible solution. Together with place cells in the hippocampus [O'Keefe and Conway, 1978], grid cells in the entorhinal cortex likely underpin the ability of animals such as rodents to model space during navigation. In particular, grid cells fire in repeating patterns as space is traversed [Hafting et al., 2005]. Using multiple grid cells of different scales and orientations, the location of an animal can be uniquely encoded [Fiete et al., 2008]. Importantly, this location representation can be updated to support path integration - that is, given information about self-movement, an agent can determine its new location by reading out from grid cell activity [Hafting et al., 2005, Moser et al., 2008]. Recent experimental evidence has uncovered the presence of grid cell-like activity in the *visual* space of primates [Killian et al., 2012, Nau et al., 2018, Julian et al., 2018]. Theoreticians have argued that grid cell-like computations might be used to build object representations in diverse sensory modalities [Hawkins et al., 2019], including vision [Bicanski and Burgess, 2019]. This is an intriguing solution to our opening problem, but the demonstration of object recognition with such computations has so far been limited to either synthetic objects [Lewis et al., 2019], or visual tasks requiring the recall of a memorized example [Bicanski and Burgess, 2019], rather than generalization to unseen examples of an object class.

Neurally-motivated systems that can solve rapid object learning and recognition given saccade-like visual inputs are therefore lacking. We set out to address this by implementing a biologically plausible network, called GridCellNet, based on cortical columns and grid cell-like computations. The system uses rapid Hebbian-style learning to associate sensed features and their spatial location in the internal reference frame of an object (token creation), while dendritic segments enable the system to encode predictive states. Locations are encoded by activity in grid cell modules that are updated with self-movement. Successive movements and the associated predictive computations enable GridCellNet to determine whether an object is present based on the spatial arrangement of features in a visual environment (conjunction detection), even when only a handful of training examples are available.

GridCellNet addresses the challenge of arbitrary sequence inputs, and due to the use of an internal reference frame for representing space, marks an important step towards sequential-input classification that is translation invariant. As the object recognition it performs is dependent on the meaningful spatial arrangement of features, GridCellNet is robust to scrambled, nonsense images. Furthermore, the system includes predictive capabilities, enabling completion of an image given partial inputs. Finally, GridCellNet's properties naturally confer rapid learning (functioning with both few training examples and few weight updates), and robustness to learning additional classes, while retaining classification performance on older classes (continual learning). We evaluate the performance of GridCellNet in these task settings, and compare it to typical machine-learning approaches. While our evaluation is limited to MNIST [Lecun et al., 1998], robustness on this simplest of visual data-sets is far from resolved [Mu and Gilmer, 2019, Schott et al., 2019, Tramèr et al., 2020a]. In accordance with human capabilities, our system outperforms more traditional machine learning approaches in the challenging setting we explore.

To summarise, our primary contributions are to:

- Implement a biologically-motivated architecture that uses arbitrary sequences of local visual features across space to learn objects and recognize them. We provide evidence that this ability is dependent on integrating self-movement for sensory predictions, which in turn confers robustness to nonsense images that carry no semantic meaning for a human.
- Demonstrate the ability of our network to successfully generalize to unseen objects given a series of sensations with arbitrary starting positions and sequence order, a necessary element for translation invariance, and a form of out-of-distribution generalization. We show that other machine learning systems, such as recurrent neural networks, struggle by comparison on limited training examples (few-shot learning).
- Demonstrate additional benefits of the proposed architecture, such as robustness under continual learning, and the ability to predict unsensed regions of the input.

2.2 Related Literature

As noted, the demonstration of object recognition using grid cells has so far been limited to either synthetic stimuli [Lewis et al., 2019], or the recall of memorized examples [Bicanski and Burgess, 2019], rather than generalization to novel samples of an object. Related work in robotics that uses sensory and self-movement information has also been limited to the recall of training examples [Pezzementi et al., 2011, Browatzki et al., 2014]. This is where our work most significantly builds on the prior state-of-the-art.

We also contrast our approach to previous methods in multi-view object recognition in machine learning. These systems often assume that there is no spatial structure to the disparate sensory inputs, instead aggregating them in an agnostic manner [Ranzato, 2014, Seeland and Mäder, 2021]. Alternative approaches that do make use of spatial information do so using an external reference frame (such as coordinates in the input space) [Larochelle and Hinton, 2010, Mnih et al., 2014, Wong and Yuille, 2015], which constrains their ability to generalize to out-of-distribution locations. By using an internal reference frame to perform inference, our approach makes use of spatial information while avoiding the limitations of relying on a fixed, external reference frame. An extended discussion of work related to our approach is provided below.

2.2.1 Grid Cells

During spatial navigation in an animal such as a rat, grid cells are notable for firing at regular intervals as the space in an environment is traversed. These points of activity correspond to a triangular lattice with a particular phase, orientation and scale [Hafting et al., 2005] (Figure 2.1a). Grid cells with the same orientation and scale, but different phases, form what are known as grid cell 'modules' [Stensola et al., 2012]. As a rodent moves, any individual grid cell's activity is ambiguous as a means of encoding the animal's position. The joint activity of multiple grid cell modules, however, can uniquely encode a position. Importantly, this encoding scheme has a large representational capacity [Fiete et al., 2008] (Figure 2.1a middle). Information about self-movement is used to update the current location representation by each grid cell's firing corresponding to the

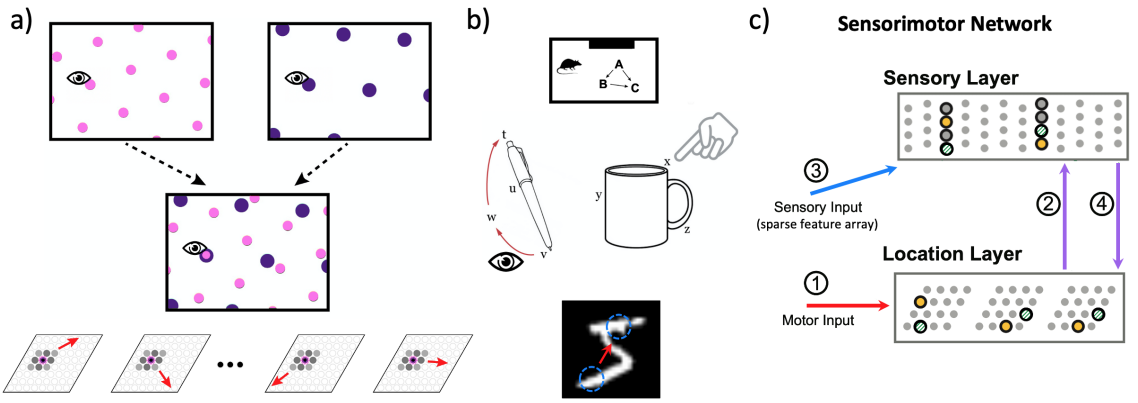


Figure 2.1: *Using grid cell representations for object recognition.* a) The combination of multiple grid cells of different scales and orientations can uniquely encode the location of a sensor (e.g. fovea). Here we use multiple grid cell modules with sparse activity (bottom, each indicated by a rhombus) to encode and update the sensor's location with self-movement information. b) It is hypothesized that this process can be used for object recognition with active sensors, and we use sequences through a discrete grid of feature patches extracted from MNIST images to test this. c) A high-level representation of the sensorimotor network that forms the basis of GridCellNet. This network takes in motor input when the sensor moves (1) and updates its location representations. The current location representation is used to predict incoming sensory information (2), before this is received (3). Correctly predicted sensory information is then used to update the location representation (4). Locations are initially ambiguous, represented using a union of locations, and disambiguated over time via sensory input. Yellow and textured green dots in the location layer indicate two different objects which are compatible with the current sequence. Classification is successful once the representation is primarily consistent with a single object class. The two-layer network is based on cells with sparse binary activity, dendritic segments, and Hebbian-like learning, following Lewis et al. [2019], and with figures reproduced/modified with permission from the authors.

positional change (Figure 2.1a bottom). This process, known as path-integration, means that after returning to the same position, the same grid cells will be active regardless of the path taken [Hafting et al., 2005, Moser et al., 2008]. Note that our work does not deal with how grid cells might actually implement path integration - rather we explore the significance of path integration in a neural population for developing useful object representations. The combined properties of a large capacity for unique spatial representations and path integration enable grid cells to act as a powerful substrate for encoding spatial information. In primates, grid cell-like activity has been observed in visual space [Killian et al., 2012, Nau et al., 2018, Julian et al., 2018]. As such, our work will explore the potential significance of grid cells for visual processes, such as object recognition given a series of saccades.

2.2.2 Cortical Models

Our work builds on previous models of the cortical architecture of the mammalian brain. Hawkins et al. [2017] demonstrated that networks with a columnar architecture, where different layers correspond to sensory and location-based representations, can learn inputs such as objects composed of synthetic features. Neurons in these layers receive external sensory and self-movement information, while they share connections that enable learned associations between features and locations, as well as predictions during inference. Neural activity (including input features) is represented in the distributed activity of sparse binary vectors - a form of encoding where the dimensionality used is relatively high, but where only a small subset of the available nodes are ever active at a time (taking values of 0 or 1). Such encoding has several appealing properties, including tolerance to noise [Ahmad and Scheinkman, 2019], a large representational capacity, and the ability to encode notions of similarity between objects [Hawkins and Ahmad, 2016].

2.2.3 Grid Cells for Object Recognition : Models

The cortical models in Hawkins et al. [2017] did not explicitly discuss how the brain might implement the encoding of location information. In Hawkins et al. [2019], it was suggested that neurons akin to grid cells might exist outside of the entorhinal cortex in cortical columns throughout the brain, and could thereby support spatial encoding in sensory modalities such as touch and vision (Figure 2.1b). Regardless of their precise location, grid cells could be used to encode feature locations in an object's own reference frame. The idea that each of the columns throughout the brain would be learning object representations in a massively parallel process was dubbed the Thousand Brains Theory. This nomenclature was to contrast the theory to those that suggest a more strict hierarchy with object-like representations only existing at certain levels of processing [Hawkins et al., 2019]. While models developed from this theory were capable of rapidly learning objects and performing recognition, this was limited to synthetic data-sets [Lewis et al., 2019], as well as the recall of training examples, rather than generalization to novel object classes.

In Bicanski and Burgess [2019], a system that relied on grid cell computations to recognize images was implemented, similar to our own approach. Importantly, however, they

explored object recall (i.e. from the training data-set) rather than generalization to unseen data. As such, the implementation assumed that at any given time-point, only one learned representation would be consistent with the observed input, and so the system only represented one possible hypothesis. An additional difference from our work to Bicanski and Burgess [2019] is that, in GridCellNet, the location representation in the grid cell layer needs to be inferred from sensory and self-movement information (Figure 2.1c). In Bicanski and Burgess [2019], all objects share a fixed reference frame across learning and inference, and so with perfect path integration, the system always knows exactly where the current fixation is in the external environment. As we discuss separately, this raises issues for translation invariance.

In addition to the limitations of generalization highlighted above, work in Lewis et al. [2019] and Bicanski and Burgess [2019] did not explore the potential benefits of a grid-cell based architecture for other tasks such as visually reconstructing unsensed regions of an image, or for challenging training settings such as continual learning. Furthermore, no comparison was made on their primary tasks to standard machine learning approaches such as recurrent neural networks (RNNs). As such, our work serves as a first systematic assessment of the potential benefits of such an internal, reference-frame based system in a machine-learning setting.

2.2.4 Grid Cells for Object Recognition : Biology

Experimental evidence for a role of grid cells in encoding visual space has been found both at the single-cell level in primates [Killian et al., 2012], and using fMRI in humans [Nau et al., 2018, Julian et al., 2018]. In both cases, these experiments focused on entorhinal cortex, although similar grid-like activity has also been observed in the pre-frontal cortex of humans [Julian et al., 2018], and there is early evidence that grid-cell responses may be represented more broadly in the brain, such as in somatosensory cortex [Long and Zhang, 2021]. It is thus plausible that grid cells would play an important role in visual processing, regardless of their precise location in the primate brain.

Regarding regions in the medial temporal lobe that share connectivity with entorhinal cortex, significant evidence exists that the perirhinal cortex plays an important role in *both*

recall memory, and general object perception [Murray et al., 2007, O'Neil et al., 2009]. For example, to distinguish objects of different classes, where this distinction cannot be made on simple features alone, Rhesus monkeys appear to require intact perirhinal cortex [Buckley et al., 2001]. Similar effects have been found in humans, where clinical lesions in perirhinal cortex are associated with difficulties in distinguishing objects, particularly when there is significant overlap in the features that compose them [Barense et al., 2005]. Imaging studies in humans have suggested a role for perirhinal cortex in both object perception where the feature overlap between objects is high, as well as recall tasks requiring the identification of a previously seen stimulus [O'Neil et al., 2009]. Importantly, the perirhinal cortex shares bi-directional connections with entorhinal cortex [Suzuki, 1996]. The hippocampus also plays an established role in declarative memory and has associations with entorhinal cortex, although its importance for the memory recall of objects (as opposed to e.g. places) is a matter of debate [Barense et al., 2005, Murray et al., 2007]. In this work, we will examine whether grid-cell computations can support both recall of previously seen objects, as well as the recognition of novel examples of an object class. We further note that the above experimental settings are similar to the one that we will explore, where *local* features of an MNIST digit carry minimal, ambiguous information about an object class.

2.2.5 Perception Through Saccades in Humans

There is a long history of studying the representations that humans develop and combine over saccades, a process known as transsaccadic integration. Despite extensive study, the precise nature of these representations remains unclear [Higgins and Rayner, 2014]. At one extreme, it is suggested that humans view the world with very little if any history of previously represented features, as suggested by phenomena such as change blindness, where humans are remarkably insensitive to changes in the image during saccades [Grimes, 1996]. At the other extreme is the suggestion that humans maintain a detailed, picture-like representation across saccades [McConkie and Rayner, 1976].

Experimental evidence supports a compromise of these views. On the one hand, humans cannot piece together percepts across saccades at a pictorial/pixel level, such as complex

arrays of random dots [Rayner and Pollatsek, 1983]. Despite this, there is evidence that integration occurs at a more abstract, feature level. For example, humans are able to integrate sequentially presented points of light to make judgements about the shape of the triangle they form [Hayhoe et al., 1991]. With regards to object recognition, humans can also recognise an object that is built up over progressive saccades, such as drawings of animals hidden among oriented lines [Holm et al., 2008]. Our work aligns with this evidence by demonstrating a model that builds up and integrates representations across space at a feature (rather than pixel) level.

It is also worth highlighting the central role that attention plays in human perception. In order to efficiently sample the environment, humans rely on signals such as bottom-up saliency to inform future fixations. There has been significant effort in the literature dedicated to modelling how the serial application of attention is driven [Borji and Itti, 2013, Wloka et al., 2018]. We note that we do not explicitly model attention in this work, such that fixation selection is described by a uniform distribution with inhibition of return, although it could be integrated to enable the system to more efficiently sample the input space.

Finally, we note that the grid-cell computations we consider will explore representing visual features with an internal, object-centred reference frame. However, this is complementary to other internal reference frames of visual space used during saccade-based vision. For example, encoding visual features and objects in a *head*-centred reference frame is likely important for processing familiar objects, and might support tasks such as sensorimotor integration and the ability to maintain stable visual perception [Mender and Stringer, 2013].

2.2.6 Multiple Views in Machine Vision

Object recognition in computer vision often focuses on processing static images from a single viewpoint. Systems that do integrate views sampled from multiple locations often make no assumptions about their spatial arrangement, simply aggregating these inputs in a spatially agnostic manner [Ranzato, 2014, Seeland and Mäder, 2021]. Other systems that do take account of spatial information typically make use of an external, ground-

truth reference frame. Such spatial information might take the form of indexed positions in the input image [Wong and Yuille, 2015], a 2-dimensional coordinate [Mnih et al., 2014, Larochelle and Hinton, 2010], or a combination of the 3-dimensional coordinate, pitch, and yaw of a camera in a room [Ali Eslami et al., 2018]. Furthermore, this spatial information is often used to solve machine vision tasks such as scene representation rather than object recognition [Ali Eslami et al., 2018]. Finally, in settings where translation invariance is explored, performance relies on training on tens of thousands of examples at different possible locations in the external environment [Mnih et al., 2014, Ranzato, 2014].

While computer vision techniques for object recognition typically focus on processing images in isolation, the field of robotics faces similar challenges (and potential benefits) to biological agents of embodiment and the ability to interact with the environment [Bohg et al., 2017]. Similar to the work we present here, Browatzki et al. [2014] presented a classification system in a robotic agent which framed visual object recognition as a problem of localization (what 'environment', i.e. object, is the sensor viewing, and where is it viewing it from). In their case, recognition was investigated in a robot rotating a 3D object, and thus identifying its location on a view sphere. The object was inferred from its sensory observations and proprioceptive data via an iterative particle filter, not unlike our system. Pezzementi et al. [2011] implemented a similar approach in a robotic system for touch perception of three-dimensional letters, and in particular demonstrated the benefit of such an approach for translation and rotation invariance. Notably, however, these approaches evaluated recognition on the same objects that were learned on (i.e. not investigating true generalization of object classes), and did not make use of grid cell representations or other biologically plausible elements when implementing the recognition algorithm.

Work in robotics has recently inspired new approaches in more general computer vision. Hoang et al. [2020] presented a classifier where, similar to capsule networks [Sabour et al., 2017] and our own work, recognition is predicated on the consistency of object features within an internal reference frame for the object. Unlike our work, they do not make use of grid cells, instead assuming an idealized 2D map for encoding feature locations,

and they train and evaluate with five objects with highly diverse features that provide locally discriminative features (e.g. a teddy bear vs. a patterned flower vase), rather than assessing generalization to novel examples of similar classes.

2.2.7 Associative Memories

While the above discussion focused on classification (our primary task), it is also worth highlighting work in machine learning and related neuroscience models on associative memories. In particular, associative memories relate to the recovery of information given partial or corrupted inputs, which can therefore be used for computations such as predicting the contents of the unsensed part of an image, a secondary task that we will consider. Similar to machine learning approaches in object recognition, these techniques typically consider a massively parallel setting, where a proportion of features are provided in a fixed reference frame (e.g. retinotopic space), such as the top half of an image. The features at unsensed regions can then be queried and potentially recovered, which might not be visible due to occlusion, or may be corrupted by noise. Architectures that can be employed in such settings include Hopfield networks [Hopfield, 1982], auto-encoders [Radhakrishnan et al., 2020], and neural networks that implement predictive coding [Salvatori et al., 2021, 2022]. We emphasize that in our setting, the model will receive features sequentially in an unspecified reference frame, and the location of the features in an internal reference frame has to be inferred *before* predictions can be made. As such, these represent different problem settings which will both manifest in an autonomous agent (such as a primate) under different conditions. In our Discussion (Section 2.6), we will therefore look at how these approaches are likely complementary to one another.

2.2.8 Few-Shot and Continual Learning

Few-shot learning is a large field, and prior work has addressed learning hand-written characters [Wong and Yuille, 2015, Lake et al., 2016, George et al., 2017], or demonstrated the benefits of memory-like mechanisms in the few-shot setting [Santoro et al., 2016]. Our intent is not to present the current work as a strong solution to the problem of few-shot learning. However, few-shot training captures our biological motivated set-

ting of humans learning rapidly from arbitrary feature sequences. As such, we use the few-shot experimental setting to evaluate the performance of our system.

In continual learning, a classifier is evaluated on its ability to learn novel classes while retaining the ability to categorize older classes. Robustness in this setting remains a significant challenge for machine learning techniques such as deep learning that display catastrophic forgetting, the phenomenon where information required to solve a novel task obliterates previously learned representations [McCloskey and Cohen, 1989]. Continual learning has received comparatively little study in recurrent architectures [Ehret et al., 2021], having first been formally evaluated by Schak and Gepperth [2019]. Such work, however, has confirmed that recurrent systems such as LSTMs suffer from significant catastrophic forgetting. Many biologically motivated approaches for continual learning have been identified for feed-forward networks, such as elastic weight consolidation [Kirkpatrick et al., 2017]. Unfortunately, recent evidence has demonstrated that these can be challenging to apply in recurrent architectures on tasks requiring substantial working memory [Ehret et al., 2021]. From the perspective of continual learning, the most similar approaches to GridCellNet are those that rely on external memory. Methods such as Gradient Episodic Memory [Lopez-Paz and Ranzato, 2017, Sodhani et al., 2020] and progressive memory banks [Asghar et al., 2020] have been applied in recurrent neural networks. Unlike GridCellNet, these approaches rely on rehearsing on a stored bank of previous experiences [Sodhani et al., 2020], or require an explicit, external memory system that continuously expands [Asghar et al., 2020]. As such, efficient and robust continual learning in sequential tasks remains an ongoing challenge.

We further note that our architecture is distinct from memory augmented networks such as Differentiable Neural Computers [Graves et al., 2016], which temporarily store information about the nodes and edges of a graph. While such a memory system can be combined with an RNN to solve challenging tasks such as inferring the relation between two nodes, this is distinct from the task setting we consider, where information must be learned and stored across *multiple* different object presentations (i.e. "graphs") for later generalization. This results in two challenges, the first of which is that any particular node (i.e. input feature) has not been seen before, as evaluation is performed on novel

object examples. The second is that any given feature is likely to be ambiguous as to the particular object that one is perceiving (e.g. a particular curved contour could be part of multiple different digits), requiring that a system first infer what "graph" it is actually on in order for subsequent processing to be meaningful.

2.3 Methods

2.3.1 Overview

Our work builds on the sensorimotor system implemented in Lewis et al. [2019], which in turn uses many of the algorithms employed in Hawkins et al. [2017]. In this chapter we address two limitations of the network in Lewis et al. [2019]. First, they used synthetic objects and features rather than those derived from natural data-sets. Second, their system was designed to only recall previously seen objects and did not generalize to other examples of an object class. In order to remove those limitations, we make two primary changes:

- We implement two sparse feature detectors, trained on images, where the extracted features approximate a series of foveations across an image. When the sensorimotor system moves to sample an image patch, the corresponding subset of sparse feature outputs is sent to the network. The same feature representations are also provided to the other sequential classifiers we compare to.
- We modify the main classification step, using multiple learned location representations as features for object recognition. Classification then operates on the set of active grid cells using a linear transformation followed by a biologically plausible normalization process. Like the core sensorimotor network, the classification layer makes use of Hebbian learning to support both rapid and continual learning. Furthermore, the use of a thresholded output enables potentially early termination of the classification response.

Code necessary to implement the models and the majority of our results is available at `https://github.com/numenta/htmpapers`.

2.3.2 Sparse Feature Extraction

The classifiers we consider, including GridCellNet, receive local features extracted from MNIST images, where these features correspond to the current spatial location of the sensor (e.g. fovea). To implement this, we perform a feature extraction stage, where this is designed to emulate the processing performed by the primate ventral stream at successive fixations, such that the networks must then make sense of the spatial arrangement of these feature snapshots. In this section, we will describe how these sparse features are extracted, before subsequently describing the core sensorimotor architecture of GridCellNet in Section 2.3.3.

In order to handle realistic images, we use two different networks to pre-process the MNIST images and thereby generate sparse binary features at multiple image locations. These two different approaches correspond to two different experimental conditions. The first of these we term the Rich Local Feature (RLF) setting, where the local features are partly diagnostic about object identity, can overlap with one another, and correspond to a reasonable portion of the entire input space (i.e. have a larger receptive field). The second of these is the Impoverished Local Feature (ILF) setting, where the local features are more limited (e.g. simple line segments), non-overlapping, and correspond to a smaller region of the input space.

In both settings, we treat the entire input space that the sequential classifiers will interact with as a discrete grid-like representation of the environment, where at any given point, the classifier receives information from one location in this grid. Note therefore that none of our classifiers (GridCellNet or the classifiers we compare to) receive direct pixel inputs as their features. To anticipate the details below, we note that the RLF setting results in a 5×5 environment of local features, while the ILF setting results in an environment of dimension 4×4. Note that although each feature in the ILF setting corresponds to a smaller region of pixel space (i.e. smaller receptive field), features at different locations do not overlap with one another, and thus the environment space is of a smaller dimension. We now explain the details of the feature extraction methods, and in particular the relevant neural networks that we utilize.

CNN Model for Generating the RLF Setting For the first feature extraction network, we trained a convolutional neural network such that it could generate sparse binary features at multiple image locations. Intuitively, this approach is intended to create a map approximating how a fovea might represent the input at different regions. The use of a CNN for this purpose is motivated by the assumption that the representations will be approximately translation equivariant, i.e. that each high-level, sparse feature is the product of approximately the same transformation, simply shifted in space. This is a reasonable assumption in CNNs due to weight sharing, particularly when they are shallow as in our case [Zhang, 2019]. Note that this pre-processed input to our downstream classifiers is therefore not treated as a retinotopic map, but rather a 2D discrete map of the entire visual environment. In the CNN, a k-Winner Take All (k-WTA) layer [Ahmad and Scheinkman, 2019] is used to enforce sparsity in the middle layers of the network, and these representations are binarized before being passed to the downstream classifiers (i.e. both GridCellNet and the classifiers we compare to).

To describe this in greater detail, we trained a convolutional neural network (CNN) [Lecun et al., 1998] in a supervised paradigm on a subset of the MNIST training data-set of handwritten digits (54,000 images) [Lecun et al., 1998], tuning it using a hold-out cross-validation section (6,000 images). This encoder network has the architecture shown in Figure 2.2a. A k-Winner Take All (k-WTA) layer [Ahmad and Scheinkman, 2019] follows the second max-pooling operation to enforce sparsity in the representation (Figure 2.2c). While the non-zero values in this layer take on real-number values (necessary for useful gradients during learning), we require binary feature vectors for input to the sensorimotor network. We therefore used the network after training to pass images through until the k-WTA layer, and then binarized this representation, providing us with a 5×5 grid of features. Each grid location is a vector of dimension 128 which contains regional information in a sparse format, where this region covers a 16×16 pixel-patch (25% of the input space including padding), and as such nearby fixations will share some overlap. The sequential provision of these features forms the input to all of our downstream classifiers.

It is possible that binarization could lead to a significant loss of information. We therefore verified that a trained decoder network can accurately reconstruct the input image from

the full feature map. In order to achieve optimal performance, it is useful for the feature vector to have a reasonably large dimension (128), number of non-zero elements (19 during training, 29 during evaluation), and high entropy (intuitively, how often does each input feature contribute to a representation across all examples). To optimize the entropy, we made use of two hyperparameters. The k-WTA's duty cycle monitors how often a unit is contributing to representations. The boosting factor biases a unit's activity to target a given duty cycle (see Ahmad and Scheinkman [2019] for details). Tuning the duty cycle and boosting factor ensures a greater number of neurons each contribute information at some point, and the representation becomes more distributed (Figure 2.2d).

In order of layers, the CNN architecture is composed of a convolution (kernel size 5, channels 64), max-pooling, convolution (kernel size 5, channels 128), max-pooling, k-WTA, and three fully connected layers (dimensions of 256, 128, and 10). k-WTA applied to the max-pooling layer is local (that is, the k-winners are determined across all channels at a given spatial location, rather than across the entire image space). This ensures that each extracted feature vector has the same sparsity. We used stochastic gradient descent with a learning rate of 0.01, momentum 0.5, batch size of 128, and 10 epochs of training.

Patch-Wise Auto-Encoder for Generating the ILF Setting As well as the RLF setting, we would like to examine the performance of GridCellNet and other classifiers when local feature information is further impoverished. To enable this, we replicate several of our results using a more limited feature-extraction stage, where the learning signal is unsupervised, taking the form of an auto-encoder that must reconstruct a pixel patch. Furthermore, we limit the receptive fields to be smaller (7×7 pixels), and with no overlap in the pixel inputs that correspond to different locations.

The network consists of three fully connected layers of dimensions 25, 128, and 128, where the middle layer is a k-WTA (i.e. sparse) hidden layer (Figure 2.2b). As with the CNN, the auto-encoder is trained on a holdout subset of the data. In particular, each MNIST image is discretized into a 4×4 grid of 7×7 pixel patches, and processing is performed on each of these. After training, pixel patches for the relevant data subsets are passed through the auto-encoder in the same manner, and the binarized outputs of the hidden layer are fed to our downstream classifiers. As before, the hidden layer is of

dimension 128, and the same entropy and duty cycle are used as for the CNN. Training consisted of 10 epochs with a learning rate of 0.001, and using the Adam optimizer [Kingma and Ba, 2015]. In sum, this results in a 4×4 grid (rather than 5×5 grid for the CNN RLF setting) of features corresponding to the possible locations of the input space, where each of these 4x4 locations is associated with a feature vector of dimension 128.

2.3.3 Sensorimotor Network

We begin with a high-level description of GridCellNet's sensorimotor network, and its inference algorithm, before describing its mathematical details. The sensorimotor network consists of two layers, one representing the sensory input, the other the location of the sensor. Cells in both layers can be either on or off, and activity in the network proceeds through a series of discrete time steps.

The sensory layer receives the input features, as well as modulatory input from the location layer. The location layer receives movement information, as well as input from the sensory layer. The connections between the sensory and location layers are modelled via dendritic segments, the small branches on biological neurons that integrate multiple synapses [Antic et al., 2010]. The architecture of the sensorimotor network is shown in Figure 2.3, while further below we clarify the inference algorithm. A dendritic segment is deemed active if there is a significant match between the sending layer's sparse activity and sparse, learned weights. This is implemented as the dot-product of sparse binary vectors, and thus the network makes use of a form of associative memory [Hopfield, 1982, Kanerva, 1993]. This match must exceed a user-set threshold, after which these dendritic segments enable a given layer's activity to predict representations in the other layer. For classification in our task, we use the term 'object' when referring to a particular instance of a hand-written digit, and 'class' when referring to the identity of the digit.

The basic intuition for inference (i.e. following learning) is that at the outset, an isolated sensory feature is likely to be ambiguous as to the nature of the object, and so the network should encode this ambiguity with a representation that corresponds to a union of all the objects compatible with this given feature. For example, a curved contour at the top of an image might represent a particular 9 or 0 (Figure 2.4a). The object-representation

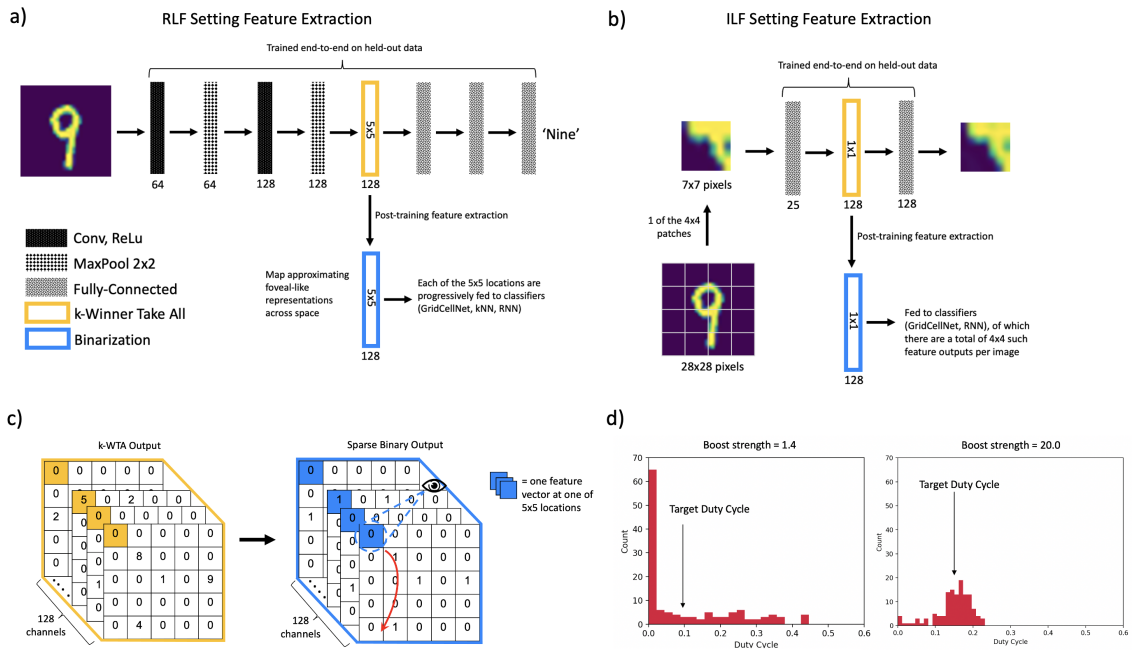


Figure 2.2: *The pre-processing neural networks.* a) The encoder CNN is trained end-to-end to perform classification, with a k-WTA operation that constrains the mid-level representations to a specific level of sparsity. Numbers below operations show the channel dimension. b) The auto-encoder network is trained to perform reconstruction of a pixel patch. c) An example of what the k-WTA and binarized representations might look like for the middle layer of the CNN. Note that in all of our tasks, the classifiers are given a series of sparse feature vectors (each of dimension 128) from 5×5 (or 4×4) total locations. This forms a sequence of sensory inputs, here represented by the eye and its movement. The order with which the features are sampled across this discrete space can either be fixed for all examples during both training and testing, or follow an arbitrary sequence. d) For optimal performance, the two parameters of k-WTA (target duty cycle and boost strength) are optimized so as to ensure most of the neurons achieve the target duty cycle. On the left is shown a typical value for the boosting factor used in past models, while on the right we show the result of using the larger boosting factor and target duty cycle that we arrived at through hyperparameter tuning.

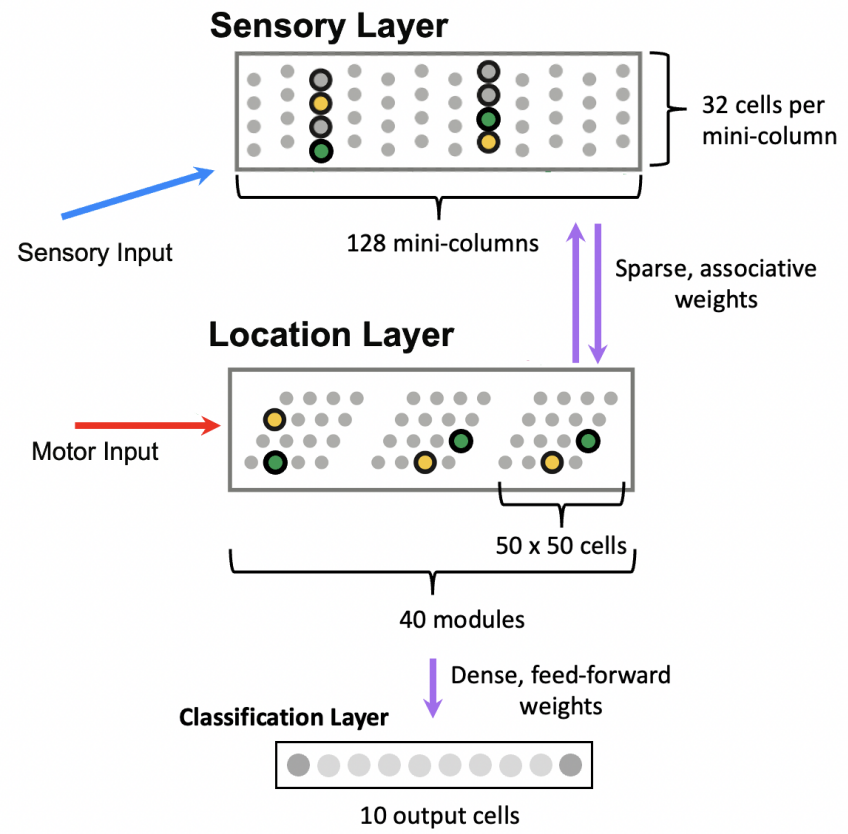


Figure 2.3: *GridCellNet's sensorimotor architecture.* Shown is a breakdown of the sensorimotor network's dimensions and connectivity. The location and sensory layers share sparse, reciprocal connections. For example, a dendritic segment on a particular sensory cell will be associated with 40 cells in the location layer, one from each module. Learning of all weights is dictated by simple, Hebbian update rules, described in the main text. The coloured nodes represent cells active during a particular stage of inference, but with yellow and green representing objects of two different classes. The bold mini-columns are those associated with the relevant dimension of the feature input. For example, if the 3rd and 8th indices of the 128-dimension feature vector are active (i.e. take a value of 1 rather than 0), then the associated mini-columns will receive input. In practice, 29 of the 128 mini-columns will receive input for any given sensation, owing to the feature-vector's sparsity. Which sparse sample of neurons in a mini-column actually become active and inhibit the others (i.e. the green and yellow neurons) is dependent on learning-based predictions from the location layer. The inference algorithm is described in greater detail separately.

is encoded by the activity in the location layer (as each learned object uses a unique location space), and so this union of multiple objects will correspond to multiple cells being active in each grid cell module. In our example, the active location representation will correspond to both where a curved contour was learned for a 9 and where it was learned for a 0 (Figure 2.4b). As additional features are sensed, the network will use its current representation of candidate objects to predict the next feature, with only those that are compatible with the subsequent sensation persisting. Notably, these predictions rely on the presence of a feature *at a given location*, and therefore this process is not equivalent to evaluating a bag-of-features. If the sensor was to move to the bottom of the image, the learned 9 representation and 0 representation will predict different features. As GridCellNet experiences more sensations, its location representation should converge to a more limited number of learned objects. The object is categorized when the representation in the location layer corresponds predominantly to a particular class, and not any others. This class-correspondence is determined by a learned linear transformation of the active grid cells, described in Section 2.3.4. We provide additional details on the sensorimotor network architecture and its implementation of Hebbian learning below.

2.3.3.1 Network Architecture

The location layer consists of 40 grid cell modules, each a lattice of 50 by 50 cells. A grid cell module has a particular scale and orientation, while the active location corresponds to the current phase of activity in the module. Grid cells can be either active or inactive, and activation is determined by either the current representation in the sensory layer, or movement applied to the previous location representation; at time step t and for grid cell module i, active cells are denoted by the binary arrays $A^{\text{loc},i}_{t,\text{sense}}$ and $A^{\text{loc},i}_{t,\text{move}}$ respectively. We model the location phase that determines $A^{\text{loc},i}_{t,\text{move}}$ using a square rather than the biologically motivated triangular lattice used in Lewis et al. [2019], although this has no major consequence for the system.

The sensory layer is identical to that used in Hawkins et al. [2017]. The input features, described in the previous section, are binary vectors of length 128 with 29 active values (i.e. approximately 77% sparsity). The sensory layer in turn consists of a corresponding 128 mini-columns, which receive the input features in a one-to-one fashion. Each mini-

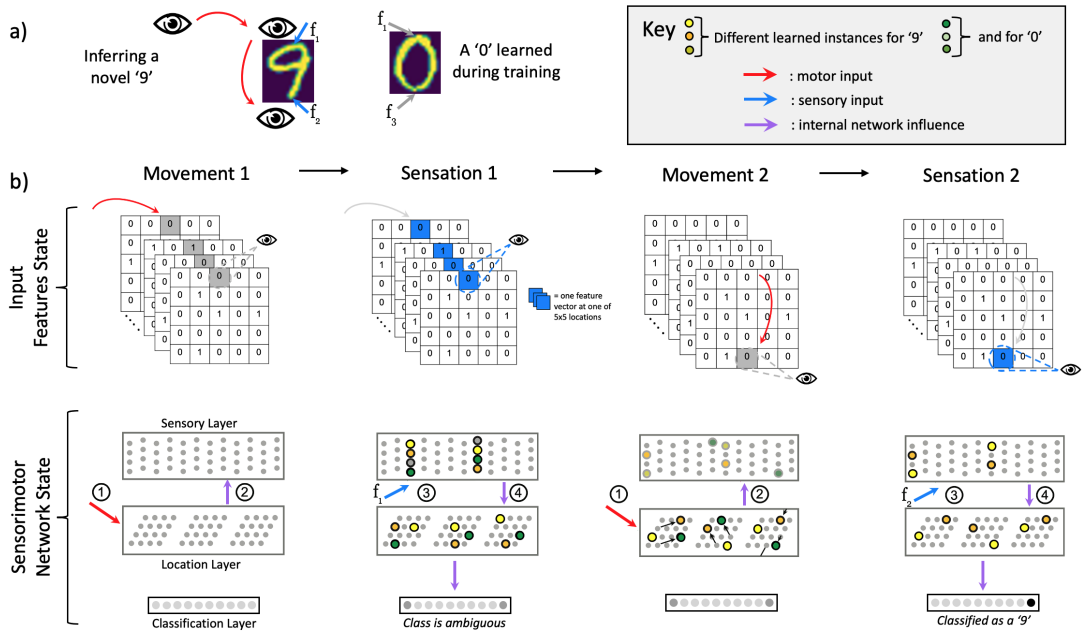


Figure 2.4: *Overview of GridCellNet's classification algorithm.* a) The algorithm's concept at a high level. A single sensation (f_1) is likely to be ambiguous as to the nature of the object. Correct inference requires integration over several features, such as a movement to the location of f_2. b) Sparse feature vectors extracted from a CNN feature map (upper row) represent an approximation of 5×5 foveations of the input. These sensations and associated movements are sequentially provided to GridCellNet's sensorimotor network (bottom row). When the sensor moves to its first location (<1>, under Movement 1), there is no location representation on which to base sensory predictions (<2>). As a result, the first sensation (f_1), activates all the cells in columns that receive input (<3>). This activity then activates location representations which are consistent with that feature input (<4>). This will be a union of multiple object representations, some of which are consistent with the target class (shades of yellow), and some of which are not (textured shades of green). As multiple class nodes receive active input, with no clear winner, the class identity of the object is ambiguous. With the next sensor movement (<1>, under Movement 2), the location representations of the grid cells are updated using path integration. The active grid cells then provide a prediction (via the modulatory impact of dendritic segments) to the sensory layer (<2>). The next sensory input (<3>) is consistent with two previously learned examples of 9's, i.e. with that feature representation at that location *given the previous feature representations at their locations*. The remaining active location representations (<4>) now drive a single target class above a threshold relative to the activity of the other class nodes. Note that Movement 1, Sensation 1, Movement 2, and Sensation 2 indicate the same two-layer network over the temporal progression of the algorithm. Figure adapted with permission from Lewis et al. [2019].

column in the sensory layer consists of multiple cells (here 32). This enables the mini-columns to use sparse activity to encode features associated with particular objects (i.e. location representations), where again, cells can be either active or inactive. The active cells in mini-column i at time-step t are denoted by the binary array $A_t^{\text{in},i}$.

We now describe the inference algorithm at the sensorimotor level in greater detail.

Stage 1: Using Movement to Update the Location Representation. If the location layer has active cells, then each module uses the current movement information to compute a new set of active cells. Recall that the currently active cells are represented by the binary activity of a square lattice of 50 by 50 cells, and that each module has a particular scale (s) and orientation (θ). Each module will apply a translation to this activation pattern, according to the movement information, which corresponds to a 2D movement vector $\overrightarrow{m}_t$. However, the actual shift ($\overrightarrow{d}_t$) in each grid cell module's lattice differs according to that module's scale and orientation. With $s = 1$ and $\theta = 0°$, this simply corresponds to a discrete shift in the representational space (e.g. $\overrightarrow{d}_t = [2, 1]^\intercal$ corresponds to 2 cells down, and one cell right). To determine the translation vector $\overrightarrow{d}_t$ for any given grid cell module, the following transformation is applied to the movement vector:

$$\overrightarrow{d}_t = M_i \overrightarrow{m}_t \tag{2.1}$$

where

$$M_i = \frac{1}{s_i} \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \tag{2.2}$$

and where i denotes the particular grid cell module. Note that if $s = 1$ and $\theta = 0°$, then M_i is simply the identity matrix. We emphasize again that, as this description demonstrates, the model assumes that path integration is implemented by some biological process; it does not seek to determine how path integration is actually implemented by the brain.

The translated 50 by 50 pattern will often not align neatly with the original 50 by 50 cells. Typically each active cell in the pattern will land on the corner between four cells, so

each active cell will activate up to four cells after the translation vector has been applied. During inference, this is indeed what happens, but during learning we allow the grid cell module to have more certainty about the current location. In particular, over the course of learning, translating the active pattern will not increase the number of active cells. Instead, the module's internal state includes a list of high-precision active phases, specified by the 2D range $[0, 1) \times [0, 1)$; the module applies the translation to those phases, rather than estimating those phases from the current set of active cells. This difference in the algorithm's behaviour in inference and learning reflects the fact that binary representations will always lead to some spatial uncertainty during inference. For a more in-depth discussion of the mathematical implementation of grid cells, we direct interested readers to Lewis et al. [2019].

Returning to the higher-level algorithm, we note that when inference *begins* for a new object, no location information is available - this will instead become available at stage 4, discussed below, and so inference proceeds to stage 2.

Stage 2: Predicting the Sensory Input with the Location Representation. The cells in the mini-columns have dendritic segments which receive activity from the location layer. Let D_c be the *set* of dendritic segments of cell c, with segments indexed by d, and where each segment has multiple synapses. If a dendritic segment is active, this is deemed to represent that the activity in the location layer is predicting that a sensory cell will become active, and the sensory cell is therefore placed in a predictive state. Let the binary vector π_t^{in} denote the sensory cells that have at least one active dendritic segment, and θ^{in} a dendritic threshold, then:

$$\pi_t^{\text{in},c} = \begin{cases} 1, \exists_d [D_{c,d}^{\text{in}} \cdot A_{t,\text{move}}^{\text{loc}} \geq \theta^{\text{in}}] \\ 0, \text{otherwise} \end{cases} \tag{2.3}$$

Stage 3: Determining Activity in the Sensory Layer. Recall that the sensory layer consists of 128 mini-columns, where each mini-column has 32 cells. Each mini-column receives active input in a one-to-one fashion from the sensory input (of dimension 128). If a given sensory cell in a column was predicted in the previous stage, and subsequently

receives activity from the input feature (that is, it is in a column receiving sensory input and was therefore correctly predicted), it will be active and inhibit any other cells in the mini-column that are not predicted. Multiple cells in any given mini-column can be active if they are predicted by the current location representation. If no cells in a mini-column are predicted but the mini-column receives sensory input, then all cells in the mini-column will become active. At a neural level, this results in predicted information being associated with lower overall activity in the network.

Stage 4: Using the Sensory Representation to Update the Location. After the sensory representation has been determined, the location layer receives inputs from the sensory layer. In particular, the sensory features help to recall location information, and supplement the location representation arrived at by path integration. Similar to equation 2.3, this is determined by the overlap between the active cells in the sensory layer, and the learned weights. In this case however, an active dendritic segment is sufficient for a location cell to now be active, such that:

$$\pi_t^{\text{loc},i,c} = \begin{cases} 1, \exists_d [D_{c,d}^{\text{loc},i} \cdot A_t^{\text{in}} \geq \theta^{\text{loc}}] \\ 0, \text{otherwise} \end{cases} \tag{2.4}$$

$$A_{t,\text{sense}}^{\text{loc},i} = \begin{cases} \pi_t^{\text{loc},i}, \left\| \pi_t^{\text{loc},i} \right\| > 0 \\ A_{t,\text{move}}^{\text{loc},i}, \text{otherwise} \end{cases} \tag{2.5}$$

At this stage, the next movement is received, and the four stages are repeated for the next time step. Classification using the location representations as input features is discussed separately in Section 2.3.4.

2.3.3.2 Learning in the Sensorimotor Network

Learning takes place via the reciprocal strengthening of connections between the active representation in the sensory layer, and the current location representation. The aim is to associate a given feature with a given location in that object's reference frame. When a new object is learned, the location representation at the first sensation is randomly initial-

ized, such that each object operates in a different location space. This enables multiple objects to be jointly represented during inference, as the probability of overlap between different objects' location spaces is low. As further sensations are performed during learning, the location representation is updated using movement information as described in Stage 1 above. For the sensory layer, as the feature-location association has yet to be learned, a random cell in each mini-column receiving input will be selected to be active. Each active cell in the sensory and location layer will then form reciprocal connections on one of their dendritic segments (d'), according to the following:

$$D^{\text{loc}}_{c,d'} := D^{\text{loc}}_{c,d'} | A^{\text{in}}_{t,\text{learn}} \tag{2.6}$$

$$D^{\text{in}}_{c,d'} := D^{\text{in}}_{c,d'} | A^{\text{loc}}_{t,\text{sense}} \tag{2.7}$$

Here "|" is used to indicate the bitwise OR operator; that is, if a synapse already exists between two cells, then it is unaffected by the learning rule.

Following the above, the network can rapidly learn objects by visiting each feature once, performing a single set of weight updates for each feature. Note that due to the path-integration performed by the grid cells, both learning and inference can take place using an arbitrary order through the features of the object - there need be no correspondence between the order taken at learning vs. that used at testing.

There are several hyperparameters of the model that might be tuned to optimize performance, such as increasing the grid cell module size to increase the capacity of the model (see Lewis et al. [2019] for a quantitative exploration of this). The main parameter we tuned was the dendritic threshold θ^{loc}. If this was too high, grid cells were too stringent in which sensory features were present to become active; if it was too low, grid cells were too easily activated by spurious sensory features. Details on how we tuned hyperparameters are provided in Section 2.3.7.

2.3.4 Classification Method

In order to enable classification of unseen objects, we extended the classification algorithm in Lewis et al. [2019]. The original algorithm required the location representation of the object to be a subset of the target representation, where the target was a single learned example. In contrast, we treat inference as the location representation driving the activity of a class node via locally learned weights. Classification takes place when the activity of a class node exceeds a user-set threshold relative to the activity of other class nodes. In short, the classification step serves as a linear classifier with the active location representations as the input features, and with weights learned via a Hebbian-like, supervised signal, rather than gradient descent. The location representation itself is determined as described in Section 2.3.3.

This approach to learning and inference is adopted so as to maintain the appealing qualities of the sensorimotor network. These consist of both rapid learning (i.e. limited weight updates), and the ability for classification to terminate early as soon as evidence converges on a given object (not unlike a drift diffusion model [Bogacz et al., 2006]). Such early convergence is more likely when the evidence for a particular object is less ambiguous (such as if it has been seen before, as opposed to a novel object), as we will see in our experiments.

This approach to classification is also appealing, as the network need never confidently "classify" a particular stimulus as a familiar object if it corresponds to something entirely out of the learned distribution. Indeed, while not something we explore here, such a lack of convergence might serve as a signal in the system to begin learning, as opposed to attempting inference.

2.3.4.1 Learning

During learning of the objects, whenever a particular location representation occurs (i.e. pattern of active grid cells), their identity is recorded. At the end of learning an object's local feature-location associations, all of the grid cells that were active during the learning of that object strengthen an associative weight with the neuron representing the class identity. This simple Hebbian-like associative learning is the only supervised signal in

the sensorimotor network. Note that these weight updates are additive. More concretely, let the binary vector α^{loc} indicate the trace (i.e. history) of all grid cells that were active at some point during the learning of a particular object. α^{loc} is a vector of length $n^2 * m$, where n is the width of a module, and m is the number of grid cell modules. W is the associative weights learned between the grid cells and the class nodes, of which there are a total of K, one for each class, such that W is a $K \times n^2 * m$ matrix. All weight values are initialized at zero. During learning, if the object being learned is of class k, and letting j index the values of α^{loc}, then the weight updates that take place are as follows:

$$W_{kj} := \begin{cases} W_{kj} + 1, \alpha_j^{\text{loc}} = 1 \\ W_{kj}, \text{ otherwise} \end{cases} \tag{2.8}$$

2.3.4.2 Inference

The system classifies an object once the activity for a given class is significantly higher than for the other candidate classes. Specifically, all active grid cells after a sensation, $\text{vec}(A^{\text{loc}}_{t,\text{sense}})$, feed on to the class nodes via the learned weights W. During inference, the activity vector of class nodes at time t is

$$v_t = W\text{vec}(A^{\text{loc}}_{t,\text{sense}}) \tag{2.9}$$

The activity of the maximally activated class node is processed via divisive normalization (a biologically plausible operation [Heeger, 1992]). Classification takes place if this value exceeds a user set threshold γ, i.e.

$$\frac{||v_t||_\infty}{\sum_k v_{k,t}} \geq \gamma \tag{2.10}$$

Classification is successful if the index of the maximally active class neuron matches the true class. Finally, to prevent occasional false confidence due to noise early in inference, the system is constrained to only perform classification when $t \geq 5$, out of a total of 25 (or 16 for the ILF setting) sensations.

2.3.5 Recall Method

As well as evaluating generalization using the approach outlined above, we also evaluate the ability of the system to recall examples from the training data-set, as this is a common evaluation method in the literature for related systems. To do so, we use the object recognition method employed in Lewis et al. [2019]. In particular, as sensations are progressively provided to the network, recall of a particular object occurs when the location representation converges to that of a single learned object. If that location is a subset of the learned location representations for the true (target) object at the corresponding position of the sensor, then recall is successful. This may fail if the representation is a subset of any other learned location representations, in which case the process has converged to an incorrect object or position. An alternative failure case is that the representation never converges to a subset of a learned representation. Formally, the recall system's probabilities for a particular object instance y during recall correspond to:

$$p(y) = \begin{cases} 1, (A_t^{\text{loc}} \subseteq L_y^m) \\ 0, \text{ otherwise} \end{cases} \tag{2.11}$$

where L_y^m is the learned location representation for any particular position m. The above recall is only considered successful if $p(y) = 1$, y is the target example object, and m is the actual position of the sensor on the object at the current time-point. Note that the position of the sensor m in the external reference frame is only used to evaluate the correctness of the system from an experimenter's point of view, and this is not information privy to GridCellNet itself.

Intuitively, recall is considered to have taken place when the representation has converged to that of a single representation. Thus, there is no additional read-out node; instead, the uniqueness of the location representation represents the object's identity. As such, this process can take place within the same system (and indeed at the same time) as the classification algorithm outlined separately. For example, the system might successfully classify an input, but never converge to a single representation, in which case no single learned example aligned well with the test object. Alternatively, it might classify an

object, and subsequently converge to a single representation. In this case, the system both recognises the class of the object, as well as the similarity of the particular test object to a previously learned example. Note that for our main result we use a slightly higher dendritic threshold θ^{loc} when evaluating recall than for generalization (20 for recall vs 16 for generalization). We discuss the significance of these processes operating in parallel and the different dendritic thresholds separately.

2.3.6 Comparison Classifiers and Decoder Network

We compare GridCellNet under various conditions to a recurrent-neural network (RNN), a k-Nearest Neighbors (k-NN) classifier [Fix and Hodges, 1989], and a multi-layer perceptron (MLP) receiving a single fixation.

RNN For our RNN, we use a long short-term memory (LSTM) classifier [Hochreiter and Schmidhuber, 1997]. This network receives an input sequence of length 25, corresponding to the 25 locations in the RLF image feature space (or 16 for the ILF setting). Each feature vector in this sequence is a sparse feature vector extracted as described in Section 2.3.2, as for GridCellNet. In addition, sensor location or movement information is concatenated to the feature, outlined below.

The RNN has a single hidden layer of dimension 128. Using additional layers did not appear helpful on cross-validation data. We used weight decay [Ng, 2004] of 0.001 and optimized with Adam [Kingma and Ba, 2015]. Learning rates were selected via a grid-search for the best performance on each classification task independently, where each task is distinguished by both the number of training examples, and whether evaluation was using a fixed or arbitrary input sequence.

Providing Movement and Location Information to the RNN In order to provide a fair comparison (and in particular the possibility of the LSTM developing an internal reference frame), we provide the LSTM with movement information similar to that provided to GridCellNet. Specifically, at each step in the input sequence, two values are concatenated to the input feature vector. These are the Euclidean magnitude of the just-performed movement, and the angle (measured in radians in standard position). These are derived from the previous and current location of the sensed feature in the 5×5 (or 4×4) input

space. This movement information is always provided to the LSTM when the input follows an arbitrary sequence. In cross-validation experiments, we also explored providing movement as the discrete x and y-displacements in the environment grid, but we found the separate magnitude and angle form to provide better classification accuracy. In any setting where the input follows a fixed sequence, the explicit location information (as the 0 to 24 index in the 5×5 space) is instead provided, as this appeared to provide better performance.

Central Fixation MLP We also compared classification to a multi-layer perceptron network that received a single fixation from an approximately central point in the image. This network consisted of a single hidden layer of dimension 256. The central points are sampled from the 3×3 central locations in the 5×5 (RLF) environment, and the 2×2 central locations in the 4×4 (ILF) environment. These locations are consistent with the moderate (but ultimately limited) ability of humans to fixate the exact centre of an object following a saccade [Foulsham and Kingstone, 2013b], as well as the fact that the centre of an object will not always represent the most diagnostic location for fixation. The network was trained to perform classification given a fixation from any of these locations, and as such, the size of its training data was effectively increased by a magnitude of $\times 9$ or $\times 4$ for the RLF and ILF settings respectively. Training was performed for 10 epochs with the Adam optimizer and a learning rate of either 0.02 (1 example per class) or 0.01 (5, 10, or 20 examples per class).

k-NN Finally, we also compared classification in our network to a k-NN classifier. This receives the same input as the RNN and GridCellNet, but as a single extended array, and without the additional movement/location information. The number of neighbours for the k-NN classifier and the dendritic threshold θ^{loc} / γ-threshold were, like the LSTM learning rates, selected via a grid-search for each classification task.

Decoder Network In order to enable visualization of the current feature representations in GridCellNet, we trained a multi-layer perceptron (decoder network) on the sparse binary features from a subset of the MNIST training data-set (54,000 images).

For the features extracted in the 5×5 environment, this decoder has a single hidden layer of dimension 512, with input $128 * 5 * 5$ and output 28×28. Learning was performed

using Adam with a learning rate of 0.001, a batch size of 64, and 10 epochs of training.

For the features extracted in the 4×4 environment, the decoder has a single hidden layer of dimension 128, with input 128 and output 7×7. Learning was also performed using Adam with a learning rate of 0.001, a batch size of 64, and 10 epochs of training.

2.3.7 Hyperparameter Selection

In Figure 2.5, we provide a breakdown of how different splits of the data are used for different steps in feature extraction, hyperparameter tuning, and learning. In Table 2.1, we list the hyperparameters arrived at for each classifier, specific to each learning task. Note that any tuning was restricted to the RLF setting; all performance in the ILF setting was a result of maintaining the same hyperparameters for each model as used in the RLF setting.

For the continual learning tasks, where fixed input sequences of 20 examples-per-class were used, the same LSTM and GridCellNet hyperparameters were used as for the equivalent standard (i.e. not continual) learning setting.

2.4 Results : Grid Cell Path Integration as a Basis for Robust Object Recognition

We now demonstrate the advantages of GridCellNet for operating in environments like those experienced by intelligent, biological agents. The settings that we consider will combine a variety of conditions, where any one of these can be challenging to satisfy. These adverse evaluation settings, in which we demonstrate GridCellNet's performance in the following results, can be summarised by the acronym SCIFFCAR:

- **S**equential : inputs to the system are fed sequentially, rather than available in a massively parallel setting. The first expectation is that this sequence should be able to follow a flexible, arbitrary path through space, rather than being constrained to always perform e.g. a raster scan across an environment. Further, a system should achieve some degree of performance given a subset of this sequence, and not require

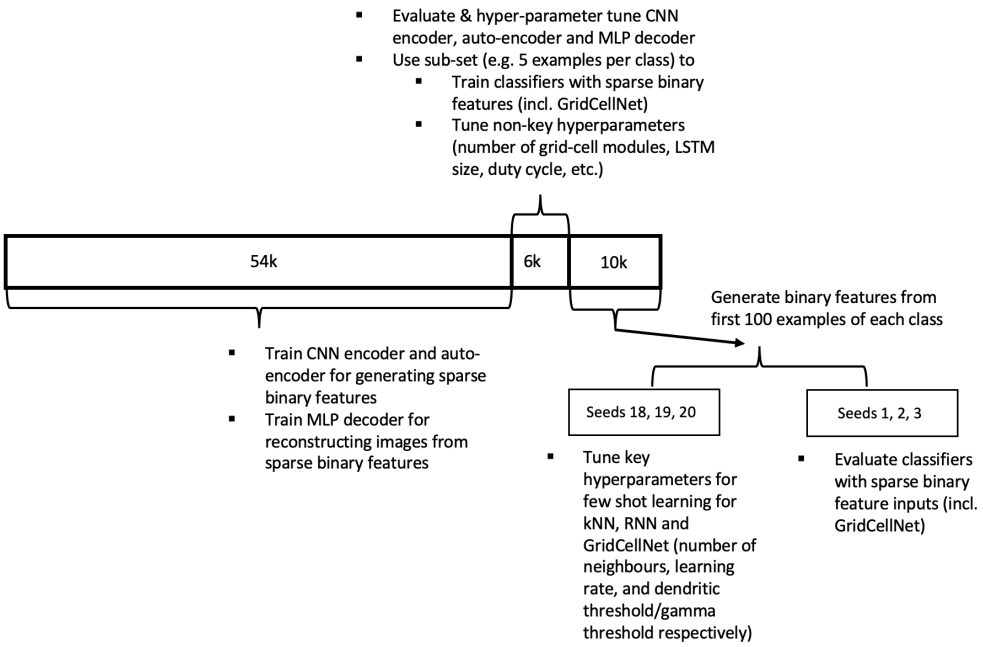


Figure 2.5: *Division of the data-set for training and evaluation.* As the downstream classifiers use features derived from a network classified end-to-end, and we are interested in few-shot learning, we divide the data-set such that the training of the various systems uses different subsets of the data. This also allows us to perform hyperparameter tuning on hold-out data. Note that although source images are re-used during the final step of hyperparameter tuning and final evaluation, the feature vectors generated by the CNN encoder will vary with the random seed used, and therefore between the settings in which hyperparameter tuning and final evaluation are performed.

Table 2.1: *Choice of Hyperparameters for Classifiers Given Fixed or Arbitrary Input Sequences.* For the spatial signal provided to the LSTM, absolute coord. refers to the precise index of the feature in the environment space; relative vec. refers to the Euclidean magnitude of the just-performed movement, and the angle (measured in radians in standard position).

# training examples per class	Fixed	Arbitrary
GridCellNet	θ^{loc}-threshold / γ-threshold	
1	12 / 0.7	12 / 0.7
5	12 / 0.5	12 / 0.5
10	14 / 0.5	14 / 0.5
20	16 / 0.3	16 / 0.3
LSTM (1 epoch)	Learning Rate / Spatial Signal	
1	0.005 / absolute coord.	0.01 / relative vec.
5	0.005 / absolute coord.	0.01 / relative vec.
10	0.005 / absolute coord.	0.01 / relative vec.
20	0.01 / absolute coord.	0.005 / relative vec.
LSTM (50 epochs)	Learning Rate / Spatial Signal	
1	0.002 / absolute coord.	0.002 / relative vec.
5	0.02 / absolute coord.	0.002 / relative vec.
10	0.005 / absolute coord.	0.002 / relative vec.
20	0.005 / absolute coord.	0.005 / relative vec.
k-NN	# of Neighbours	# of Neighbours
1	1	1
5	1	1
10	1	7
20	1	9

the entire input before useful representations emerge.

- **C**ontinual learning : it is expected that the system should be able to accommodate learning novel object classes without needing to return to, and rehearse, previously learned objects.
- **I**nternal reference frame : the input features can be associated with internally available information (in particular, self-movement), but no ground-truth information about the positions of features in the external environment should be required for tasks to be solved.
- **F**ew-shot learning : the number of training examples available is limited to less than a couple dozen per object class.
- **F**orm-specific : whilst satisfying the other conditions of SCIFFCAR, the system

should not resort to a bag-of-features approach that makes it insensitive to the spatial arrangement of features. Thus, an object should only be recognised as such if the spatial arrangement of features actually corresponds to the object in question.

- **C**lassification : the system should be able to recognise a novel instance of an object as belonging to a particular class, thereby generalizing beyond the training distribution.
- **A**ssociative memories : in addition to classification, the system should be able to make predictions about unsensed regions of the input, and recall a particular training example given a corrupted version. This should not necessitate retraining the system on establishing associative memories as a separate task from classification.
- **R**apid learning and inference : the number of weight updates required for learning should be limited (e.g. a single update of Hebbian weights), as should network changes required for inference (e.g. the reinstantation of a set of cells given a particular input), such that the system could plausibly operate on the time scales demonstrated by humans performing tasks similar to the ones we consider [Króliczak et al., 2003, Orlov and Zohary, 2018]. Note that this is a separate (potentially compounding) problem to few-shot learning, which concerns the number of training examples available, rather than the time spent learning from these examples.

2.4.1 Inference Given Arbitrary Sequences, and the Problem of Translation Invariance

The following results demonstrate that GridCellNet can perform categorization that requires generalizing to novel examples of MNIST, i.e. images that it has not seen in the training data, thereby demonstrating capabilities beyond prior related work [Pezzementi et al., 2011, Browatzki et al., 2014, Lewis et al., 2019, Bicanski and Burgess, 2019, Hoang et al., 2020]. More importantly, however, the results demonstrate classification in the setting of arbitrary sequence inputs, and while using an internal reference frame. We begin by framing the significance of a classifier that is able to handle such saccade-like inputs that follow an arbitrary path across space.

Vision in embodied agents, such as primates, faces the reality that a stimulus often cannot be sampled in its entirety through a single fixation, and so some mechanism must exist for integrating multiple foveal inputs. As shown in Figure 2.6a, a particular sequence of eye movements will be followed the first time an object is seen. The next time this object is observed, however, there is little-to-no guarantee that the eyes will sample it by beginning at the exact same location they did the first time it was seen, nor that they will subsequently follow the same sequence after this. Such an approach would require a priori knowledge about what the object is before it has been inferred, or that bottom-up saliency signals could support such a perfect alignment of saccades within the object. Experimental evidence supports such an approach being implausible. While humans occasionally demonstrate repeated patterns in the sequences they take when viewing complex objects or scenes [Noton and Stark, 1971], evidence does not support regimented, fixed adherence to such sampling being necessary nor beneficial for recognition [Foulsham and Kingstone, 2013a].

As the stimulus can move in the real world, and the starting position of where classification begins can therefore vary (i.e. translate), translation invariance represents a considerable challenge for sequential classifiers. If the classifier requires a fixed spatiotemporal input sequence regardless of where it begins on the object, it will, by definition, generally fail to sample the object (Figure 2.6b). Instead, it must be capable of beginning anywhere on the object, and subsequently follow whatever trajectory samples the input (Figure 2.6c). We note that this form of translation invariance is different to the concept explored in massively parallel processing, such as invariant recognition of an object presented to a novel part of the retina [Cox and DiCarlo, 2008, Bowers et al., 2016, Han et al., 2020, Blything et al., 2021, Biscione and Bowers, 2021], and yet it is still vital for a system that can operate robustly in the real world [Larochelle and Hinton, 2010].

Given the above, two requirements for translation invariance in a sequential classifier are i) the ability to integrate features from arbitrary starting positions and sequence inputs, ii) the use of an internal reference frame for classification. A classifier can be envisioned that can handle arbitrary sequence inputs, but whose classification process relies on an external reference frame to encode the spatial relations of features. While this satisfies

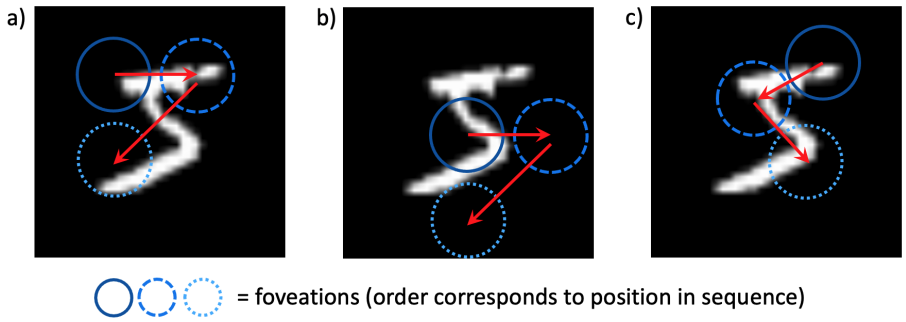


Figure 2.6: *Translation Invariance in Sequential Classifiers.* a) A sequence of hypothetical foveations during learning. b) During inference, the initial sensation may begin on a different part of the object (translation). Classifiers with a rigid input sequence requirement won't adequately sample the image. c) Regardless of where inference begins on the object, the system should sample the object and correctly classify it (i.e. translation invariance).

(i), this system will fail if that object is moved in the environment, exposing it to out-of-distribution spatial coordinates.

To achieve translation invariance, a final, third condition is required, which can be best understood with our conceptualization of the visual environment as a 2D discrete grid of foveations. In such a setting, iii) feature representations should be invariant given small translations of an object, and *equivariant* following larger translations. In other words, given a shift of a feature in the visual environment, this should result in the same activation at the neural level, but shifted by the same amount in the 2D discrete grid of foveations. Condition (iii) strictly holds in only limited cases with our pre-processed features (e.g. if the object is translated by exactly 7 pixels in the ILF 4×4 visual environment), and so we emphasize that we do not directly evaluate translation invariance, such as by moving an MNIST image in pixel space. However, in the following results, we provide empirical evidence that GridCellNet satisfies condition (i), while it satisfies requirement (ii) by definition. In particular, no information about the absolute position of features in the external world is provided, and instead GridCellNet relies entirely on sensory inputs and self-movement information. Our results therefore demonstrate the importance of GridCellNet as a step towards translation invariant sequential classifiers.

We now return to our particular classification task, where we will begin by evaluating performance in the Rich Local Feature (RLF) setting. In the first condition we consider,

the input sequence follows an (implausibly) fixed order. In particular, the features are processed following a sequential order that is the same for all objects during both training and evaluation. The other condition we consider is with arbitrary starting positions and sequences, where the input order is randomly determined for every object presentation, and is not fixed between training and testing. In the context of learning, GridCellNet performs only a single weight update per feature of an object. As such, we compare to LSTMs with both 1 epoch of training as well as 50 epochs. Note therefore that our few-shot setting considers not only exposure to a limited number of training examples, but also limited opportunity for weight updates with each training example.

For the first result, we evaluate the classifiers on the standard task of learning and generalization given a fixed input sequence from the RLF environment. As expected, all of the classifiers do reasonably well (Figure 2.7a), and demonstrate improved performance as more training examples are given. The primary exception to competitive performance is the LSTM constrained to only one weight update per set of input features. GridCellNet's learning takes place using rapid Hebbian style weight updates, and so unlike the LSTM, it can form robust representations despite only having observed each training object once. This is in contrast to the 50 epoch LTSM's rehearsing on a given object multiple times.

We next assess performance in the setting of arbitrary starting positions and sequence inputs. As predicted, only GridCellNet maintains its performance (Figure 2.7b). Despite also being given self-movement information, the LSTM does not attain the same performance as GridCellNet in the few-shot setting.

Finally, we determine the performance of the competitive classifiers (GridCellNet, 50-epoch LSTM, and single-central fixation), but in the Impoverished Local Feature (ILF) visual environment with arbitrary input sequences. As expected, all classifiers observe a drop in performance (note the change in the y-axis range), but GridCellNet continues to demonstrate its superior ability to accommodate this challenging classification setting (Figure 2.7c).

GridCellNet achieves its robustness by use of the path integration properties of grid cells, enabling the network to represent the spatial location of features in a manner that can handle arbitrary and previously unseen movements through space. After perceiving a

feature at a given location, path integration uses self-movement to meaningfully update the active location representation. This in turn predicts the learned features at the new location. The particularities of the path that was taken are irrelevant to this process, and so the system is robust to arbitrary feature sequences.

This ability to handle arbitrary sequence paths can be viewed as a form of out-of-distribution generalization. Specifically, any particular path is astronomically unlikely to be experienced more than once, and therefore will not transfer from the training distribution to the evaluation distribution. With 25 total locations, there are a total of $25!$ possible feature sequences, or around 15 million billion billion. Even assuming that 10 of 25 sensations is sufficient for classification, this represents $25!/15!$ possible sequences, or around 12 trillion. Thus, with any reasonable amount of training data, the classifier cannot rely on having previously observed a particular path for a particular object class.

We highlight that GridCellNet's inference operates entirely on sensory inputs and self-movement information (i.e. an internal reference frame). On the first sensation, the network initializes an internal representation based on previously learned objects, and uses subsequent movements in their reference frames to iteratively narrow down the candidate objects. As a consequence, the starting position of inference is irrelevant. If the entire stimulus was shifted in an external coordinate frame, such as the absolute location in a room, inference would be unaffected and as such, GridCellNet's classification system is an important step towards enabling online translation invariance in sequential classifiers. Figure 2.8 provides an example of the active grid-cells in two modules during inference of a particular object (e.g. a '9'), demonstrating how activity becomes progressively more sparse as convergence takes place.

For a fair comparison in the arbitrary-sequence experiments, our LSTM receives self-movement information, such that it could in principle also develop an internal reference frame. As our results demonstrate however, only GridCellNet has the inductive biases to perform object recognition given arbitrary input sequences given so few training examples.

We also include a k-NN model in our results. We note that the k-NN actually performs better than the LSTM with 50 epochs of training, despite the LSTM being provided with

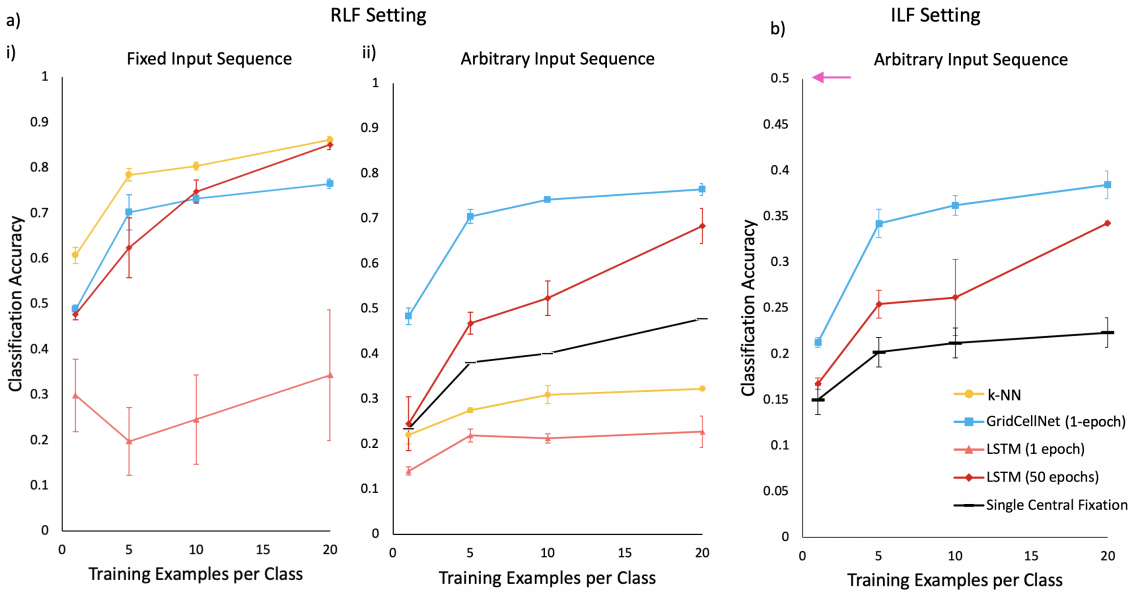


Figure 2.7: *Performance of Classifiers Given Fixed or Arbitrary Sequences of Input.* Classification accuracy on 1000 examples of the MNIST test set as a function of the number of training examples per class. In (a) and (b), performance is evaluated in the Rich Local Feature (RLF) setting, while in (c), classification is in the Impoverished Local Feature (ILF) setting. a) Accuracy when an identical sequence of passes over the input space is used for both training and inference. b) As in (a), but performance when the starting point and subsequent sequence over the input space can be arbitrary (i.e. different) between training and inference. c) As in (b), but where features are extracted from the ILF visual environment. Note the change in the y-axis limit. Error bars show the 95% confidence interval of the mean across three random seeds.

location information. This appears to be due to the challenge the LSTM faces of learning longer range dependencies given so few training examples, in spite of being provided with information about where the current feature is located in the sequence. The ability of the LSTM to solve this appears sensitive to adjustments in hyperparameters including the learning rate.

An alternative strategy in the face of inputs following an arbitrary path across space is to simply rely on a single, approximately central fixation. We include a multi-layer perceptron trained on such a task, where the fixations during training and classification fall on a random location in the central 3×3 (for the 5×5 RLF environment) or 2×2 (for the 4×4 ILF environment) positions. Such an approximate, albeit imperfect central fixation is consistent with the precision of human saccades [Foulsham and Kingstone, 2013b], and the reality that the central point of an object will not always be the most diagnostic location to fixate. We note that during training, the classifier is eventually presented with all of these possible central locations, effectively increasing the size of the training data-set. As expected, while some information can be gleaned from such a single fixation, it is inferior to integrating information over multiple image regions, even when the receptive field is relatively large, as in the RLF visual environment. Furthermore, such an approach does not address the requirement to only positively classify stimuli that correspond to the spatial arrangement of an actual object, as we will explore further in Section 2.4.3.

It is worth discussing the difficulty of the task in the ILF visual environment. In particular, less than 40% accuracy on "MNIST" might seem concerning given the simplicity of this data-set as it is normally approached (indeed when it was introduced, classifiers at the time could achieve greater than 98% accuracy [Lecun et al., 1998]). It is important however to put this result into context.

We first emphasize that our problem setting is a marked departure from the standard MNIST task, and in particular the requirements of SCIFFCAR that we consider here, including few-shot learning and sequential inputs without external spatial queues, make this a significantly more challenging classification task. This is consistent with the highly degraded performance of the alternative architectures we compare to.

We also highlight that, given sufficiently impoverished features, this is a challenging ob-

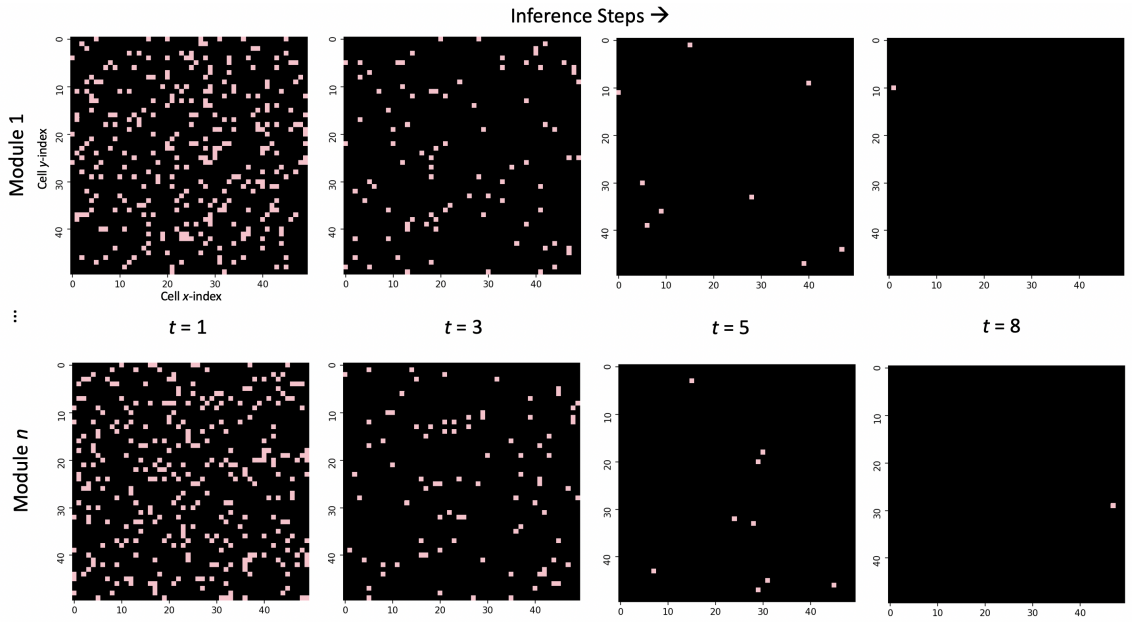


Figure 2.8: *Sample grid-cell activity during inference.* Shown is an example of the progressive change in grid-cell activity that takes place during inference. The rows Module 1 and Module n each correspond to a single grid-cell module (50×50 cells) over time; note that this is therefore not visualising the activity of a single grid-cell across space, which naturally forms a grid-like lattice. At the beginning of inference, the input feature will be consistent with many learned associations between features and locations, and so a significant number of grid cells in each module will become active. Following each subsequent movement, the sparse activity in the sensory layer will selectively activate the grid-cells consistent with the history of inputs. Over time, the activity therefore converges to only those cells that are consistent with a learned sensorimotor sequence. By the 5th sensation in this example, it might be possible to classify the object if the remaining grid cells were predominantly active for a particular class during learning (e.g. examples of 9's). If a stimulus closely matches the learned representation (e.g. is a familiar object example viewed again) the system may eventually converge to the exact set of grid-cells that were active during learning, here seen at the 8th sensation. This is akin to recalling a particular learned object example, discussed in Section 2.5.1. Note that the high precision of the grid cells is partly a machine-learning choice for improved performance on the task at hand, and partly a reflection of the limitation of a binary code. Were we to use e.g. a firing rate code, then more realistic population-sizes and sparsity for each module could likely be leveraged.

ject classification setting even for humans. For example, in Holm et al. [2008], participants had to identify animals composed of local line segments, which in spite of extensive viewing times, was associated with only 51% accuracy (chance performance 5%). Similarly, in Króliczak et al. [2003], participants had to recognize the familiarity of a previously seen object, where the object was both learned and evaluated through a manually-controlled aperture. This aperture enabled approximately 18% of the object to be viewed at any given time. In such a setting, accuracy was less than 75% (chance 50%). We note that in the ILF setting, only 1/16 of the object is visible at any time, corresponding to less than 7% of the object, and chance performance corresponds to 10%. Finally, in Foulsham and Kingstone [2013a], participants were instructed to learn a scene while provided with a sequential set of visual samples. During a recall task where they had to determine whether the scene was familiar or not, participants responded correctly on 69% of trials (chance 50%). Naturally, performance under such settings will ultimately depend on precisely how ambiguous vs. informative local features are, as well as other task factors. These results demonstrate, however, that as local features become increasingly restricted in their information, even humans find such tasks difficult.

As a final point, we note that given sufficient training time and examples, the LSTM's performance steadily improves. Deep learning architectures are undeniably powerful; indeed we used them to perform the initial feature extraction step for all of our classifiers. It is therefore likely that, given enough training examples, the LSTM's performance would match or exceed GridCellNet. Moreover, given appropriate training conditions, recurrent architectures can learn to develop grid cell-like units capable of path integration [Whittington et al., 2020]. The purpose of GridCellNet is to demonstrate the benefit of the inductive bias that path integration together with only simple, Hebbian-like learning rules can provide. The contrast in performance in the few-shot setting supports the proposed architecture as a principled and biologically plausible mechanism by which humans might rapidly learn under the challenging settings explored here, and a useful approach for designing machine learning models.

2.4.2 Rapid Inference With Partial Input Sequences

In principle, GridCellNet can successfully classify an object before it has received a complete sequence of all the features in an environment. Specifically, classification occurs as soon as GridCellNet's location representation drives a particular class node's activation above a relative threshold. Such early inference is obviously desirable as a means of increasing the efficiency of an agent relying on object classification to interact with the world.

To assess the efficiency of GridCellNet's inference process, we simply determined the cumulative accuracy as a function of the number of sensations used (total possible of 25 for the RLF setting, and 16 for the ILF setting). Note that, as detailed in the methods section, GridCellNet accumulates information for the first several sensations, but only begins attempting classification on the 5th sensation in order to avoid spurious classification. In Figure 2.9, we show that GridCellNet classifies most of the examples given to it after observing only a fraction of the total input sequence. Results are shown both from the RLF visual environment, and the ILF environment. Note that in both cases, the classifiers surpass 50% of their ceiling performance two sensations after classification is enabled (i.e. after 7 sensations total). The downstream benefits of such early convergence will be considered in later sections.

2.4.3 The Use of Self-Movement Information and Form Robustness

We have argued that GridCellNet is able to perform well in the setting of arbitrary sequence inputs because of its use of path integration. By updating internal location representations with self-movement and sensory inputs, the mutual consistency between these two sources of information enables it to accurately identify an input stimulus, even when the sequence of inputs follows an arbitrary sequence across space. It is possible however that GridCellNet is simply an effective 'bag-of-features' classifier - that classification would be successful given any random order of sensory inputs, with no alignment between their location on the object and the movement of sensors. Moreover, it is notable that the LSTM begins to approach the performance of GridCellNet on arbitrary sequence inputs as the number of training examples increases. While this is impressive, the LSTM

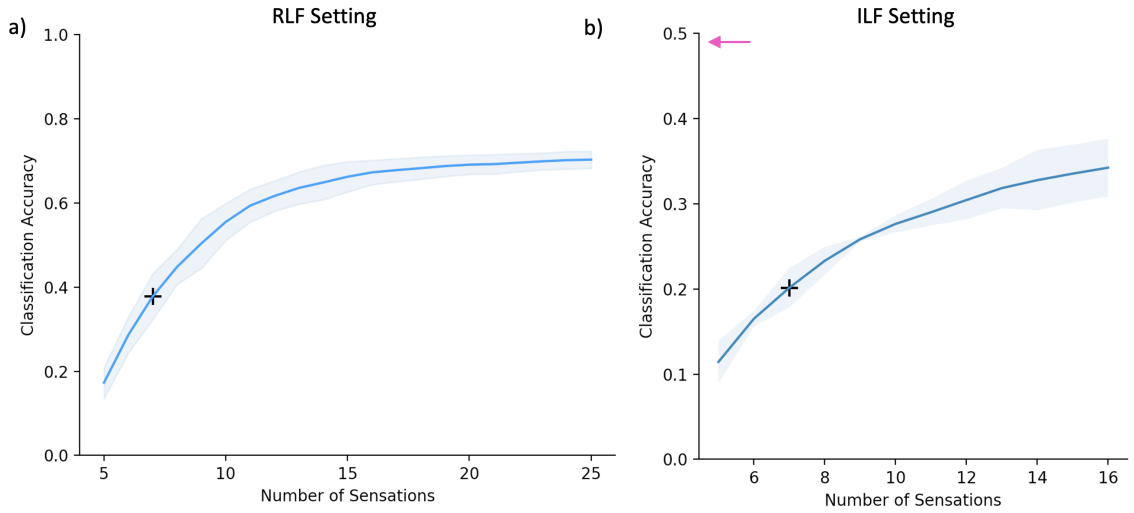


Figure 2.9: *Performance of GridCellNet as a Function of Sensations.* Accuracy as a function of the number of sensations. In both settings, GridCellNet was trained with 5 training examples per class, using arbitrary input sequences at training and test time. NB that GridCellNet waits for 5 sensations before it begins making predictions so as to avoid false hits in early inference. Results are shown when features are extracted from the Rich Local Feature (RLF) environment (a), and the Impoverished Local Feature (ILF) environment (b). The markers indicate when GridCellNet has surpassed 50% of its ceiling performance in a particular setting (both at the 7th sensation). Note the change in the y-axis limit in (b). The shaded regions show the 95% confidence interval of the mean across three random seeds.

may also be simply integrating features without any consideration of their spatial arrangement.

If either GridCellNet or the LSTM adopts such an approach, this would be concerning, as we are interested in robust object recognition that depends on the structured relations of features in space. Departure from this would make a classifier vulnerable to positively classifying distorted or spatially inconsistent visual inputs that a human would not consider to be consistent with a particular object.

To evaluate the classification method used by GridCellNet and the LSTM, we take two approaches, which are relevant for the two environments we consider. In the RLF environment, feature patches can share overlapping information, and so while it is not possible to create a single MNIST image where the feature arrangements are scrambled, we can evaluate the spatial sensitivity of the models by providing incorrect motor information during inference. In the ILF visual environment, each feature patch is entirely self-contained, and so we can generate scrambled MNIST images, where it is obvious to a human that the spatial structure has been destroyed.

The RLF Setting and False Motor Information To assess the spatial sensitivity of our models in the RLF setting, we perform a form of ablation where the GridCellNet and LSTM classifiers are provided with sensory and self-movement information as usual during learning, but then false motor information during inference. This motor information is derived from an alternative, fabricated sequence of movements over the object, and therefore does not align with the sequence of sensory inputs (Figure 2.10a). For a classifier that relies on sensorimotor alignment to predict the next sensation and appropriately update its internal representation, such an ablation should significantly affect classification accuracy. For a system that simply keys off the sensory features it receives, with no concern for their spatial relations, such a change should have minimal effect. We perform the evaluation in the setting of arbitrary sequence inputs with 10 learning examples per class.

The results in Figure 2.10b show that GridCellNet's classification accuracy suffers considerably in the context of false-movement information, while the LSTM's does not. This result is consistent with the proposal that GridCellNet has the inductive bias to make use

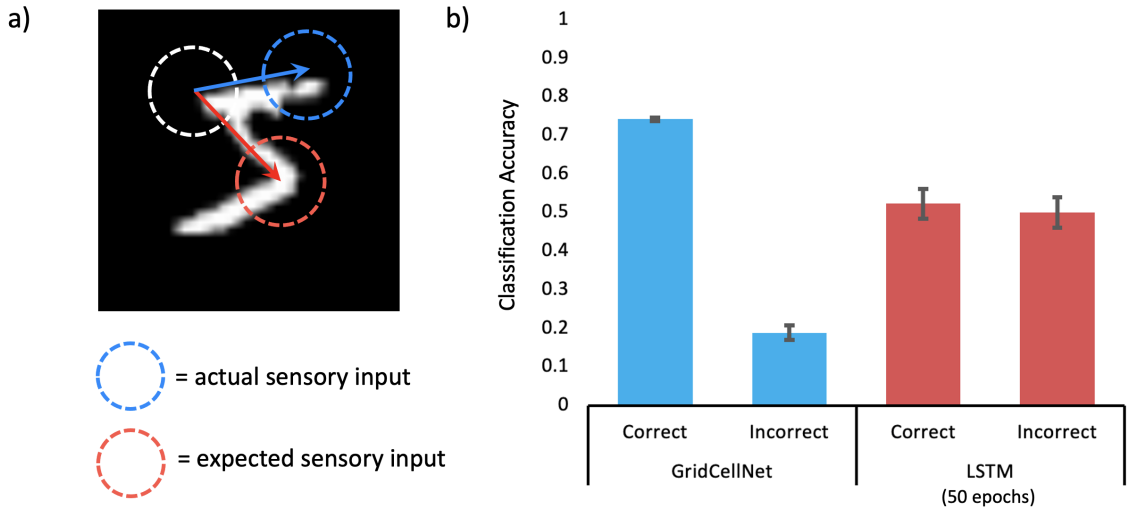


Figure 2.10: *Classifier Use of Self-Movement Information.* a) A diagrammatic representation of the ablation. We aim to measure whether the classifiers are truly integrating self-movement information with sensory information, or simply keying off randomly arranged sensations without any consideration for spatial arrangement. In this condition, the motor information provided (red arrow) to the classifiers does not correspond to the actual movement required for the provided sensory input (blue circle). b) Accuracy when given the correct self-movement information, or fabricated (incorrect) self-movement information during evaluation. The error bars show the 95% confidence interval of the mean across three random seeds.

of self-movement information, and does not simply key off a random assortment of the features that it receives.

The ILF Setting and Scrambled Images In this setting, we can perform a more intuitive intervention, where we randomly re-arrange the 4×4 grid of 7×7 pixel patches associated with any given MNIST image. Examples of both the original and scrambled images are shown in Figure 2.11a, where it is clear that the distorted images hold little, if any semantic meaning to a human. These stimuli can thus be viewed as a form of "rubbish" examples [Goodfellow et al., 2015, Nguyen et al., 2015]. As in the ablation in the RLF setting, classifiers are trained on normal, undistorted images, and then presented with the scrambled images at test time. Once again, we would expect a robust classifier that makes use of the spatial relations amongst features to observe a significant degradation in performance. This is indeed observed in the case of GridCellNet (Figure 2.11b), where the error rate is consistently within the range of chance (90%), in keeping with the lack of semantic information in the scrambled images. Interestingly, we observe that the LSTM's

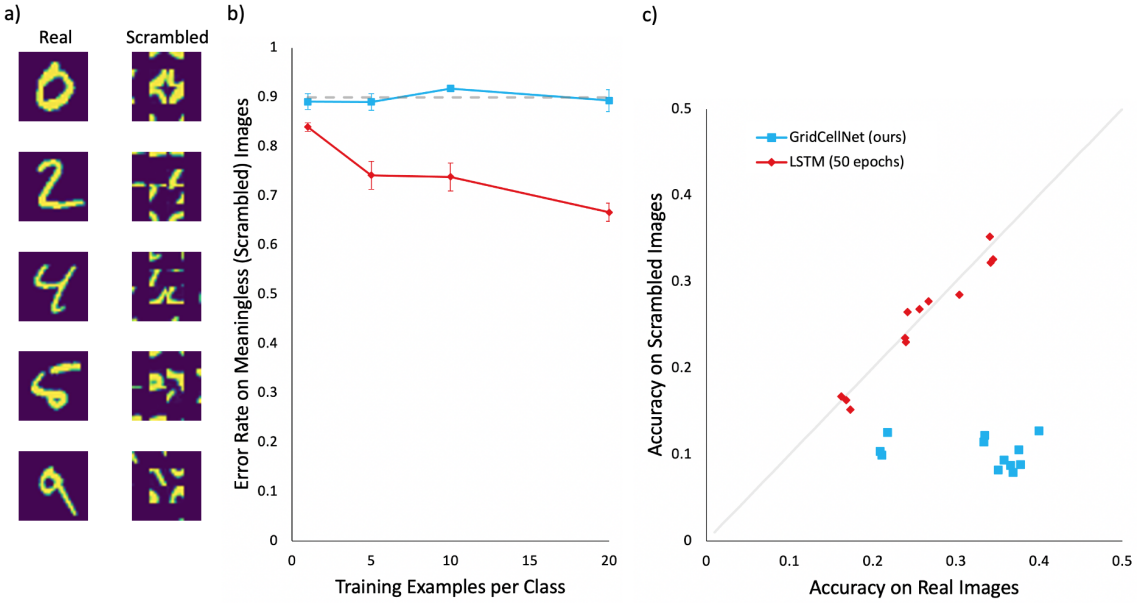


Figure 2.11: *Processing of Scrambled (Nonsense) Objects.* a) We evaluate classifier performance of the LSTM and GridCellNet in the ILF setting, where objects were either standard MNIST digits, or meaningless re-arrangements of pixel patches (scrambled). b) The error rate of the classifiers on scrambled digits, shown as a function of the amount of training data they receive. The dashed grey line represents chance. c) A break-down of the individual trained classifiers (i.e. across various amounts of training data), plotted as a function of their accuracy on scrambled objects, vs. true objects. The grey line represents unity.

error rate gradually decreases as a function of the training size, suggesting that the majority of the improvement seen on the main task has no relationship to improved processing of the spatial relations amongst features, but rests entirely on extracting increasing information from local features. This is concerning, however, as these local features contain little meaningful information to a human observer, and leave the LSTM vulnerable to the false recognition of meaningless stimuli.

Consistent with this, we observe an almost perfect relationship between the performance of an LSTM on the original task, vs. on the scrambled images (Figure 2.11c). This effect bears similarity to a phenomenon in typical machine learning classifiers, where strong performance in standard object recognition tasks [Tsipras et al., 2019], or invariance to diverse perturbations [Kamath et al., 2021], is often offset by vulnerability to adversarial examples. This observation is also in contrast to GridCellNet, where performance improvements are not associated with enhanced vulnerability to meaningless stimuli.

Primate Processing of Spatially Unstructured Objects We note that, while the simplistic approach adopted by the LSTM might seemingly resolve the task at hand given sufficient training data, it would be inconsistent with how humans recognize and understand objects (i.e. relying on global shape and the structured arrangements of features [Baker et al., 2018]). How exactly do humans perform when presented with spatially unstructured objects that are perceived in a sequential manner? Even a subtle degradation in spatial information about the arrangement of object features can impact human performance [Króliczak et al., 2003], although the size of the effect is, as in our network evaluations, dependent on the amount of information available in local features.

In particular, given sufficiently rich local feature information, humans can recognise scrambled objects above chance, even if performance is degraded in comparison to when the features share a meaningful spatial arrangement. For example, in Orlov and Zohary [2018], participants first learned and were then required to recognise novel objects when viewing these through a moving aperture. This aperture was restricted to 10% of an object at any given time. In the first condition, the rectangular aperture moved over the objects in a spatially coherent manner, and accuracy was at around 65% (chance 11%) (additional data from Orlov and Zohary [2018] provided by T. Orlov, personal communication, March 2022). Presentations which corresponded to a scrambled arrangement of the features caused a drop in accuracy to around 44%. Thus, distorting spatial relations resulted in a considerable drop in performance, although accuracy was still well above chance, consistent with the view that local features were relatively discriminatory. GridCellNet demonstrates a similar phenomenon in the RLF setting, where accuracy in spite of false motor information is 19% ($\pm$2% 95% CI, 10 training examples per class) increasing to 25% ($\pm$1% 95% CI, 20 training examples per class), in comparison to chance performance of 10%.

In Holm et al. [2008] however, local features were much more impoverished (small line segments), and they observed that no participant reported the presence of an object in the control region of the image, even though this region represented a largely analogous (albeit spatially unstructured) region to the one where an object was actually present (L. Holm, personal communication, March 2022). In other words, given sufficiently impov-

erished features, humans do not perceive illusory objects if these features do not share some underlying spatial structure. The same appears to hold for GridCellNet, where in the ILF setting, accuracy was persistently in the range of chance (Figure 2.11b).

We further note that the performance of GridCellNet in the setting of false motor information or scrambled images is consistent with the known neuroanatomy of the primate visual system. In particular, lesions to the perirhinal cortex in primates [Buckley et al., 2001] and more extensively to the medial temporal lobe in humans [Barense et al., 2005] are associated with deficits in distinguishing objects that share significant levels of feature overlap with one another, while processing objects that can be distinguished on the basis of simpler features is unaffected.

Backpropagation and Feature Use While the results in the above two experiments demonstrate the robustness of GridCellNet's use of spatial information, they also demonstrate that the LSTM uses the capabilities of back-propagation of error to learn a relatively effective mapping between randomly ordered sensory features and the output class, with no concern for their spatial arrangement. In our Discussion (Section 2.6), we will return to the strengths of such learning algorithms, and how hybrid systems are likely to provide the strongest approach.

As an aside, we once again note that we do not argue that RNNs are incapable of learning path integration, as given an appropriate architecture and learning setting, grid cell-like representations have been observed to emerge [Whittington et al., 2020]. Rather, our aim is to show that at the outset, GridCellNet has this capability, and that this enables it to process the input sequence in such a way that accounts for the consistent spatial arrangement of features.

2.5 Results : Additional Capabilities of GridCellNet

While the above results focused on object recognition in GridCellNet, we now explore additional capabilities of the architecture.

2.5.1 Recall of Training Examples

In the Section 2.4.1 we demonstrated that GridCellNet can generalize to novel examples of MNIST that it has not seen in the training data. The following section demonstrates that generalization occurs without the loss of recall ability. In particular, GridCellNet can also operate in a regime focused on recalling a specific learned example, rather than classification. This has clear relevance in the natural world, where it can be necessary to recognise whether one has seen a particular instance of an object before, and distinguish it from other similar examples.

Demonstrating recall in our model is also useful, as this is the evaluation metric (as opposed to generalization) that has been used in previous related work [Pezzementi et al., 2011, Browatzki et al., 2014, Lewis et al., 2019, Bicanski and Burgess, 2019, Hoang et al., 2020]. We note that while the following results thereby replicate the functionality seen in previous models, this is performed using an internal object reference frame. Prior biologically plausible approaches assumed a fixed reference frame across all objects [Bicanski and Burgess, 2019], rather than inferring locations from the combination of sensory inputs and self-movement. As previously noted, the use of an internal reference frame is an important condition for a sequential algorithm that can operate in a translation invariant manner.

2.5.1.1 Recall Despite Noise

Similar to Bicanski and Burgess [2019], where 33 faces, 33 objects, and 33 scenes were used, we evaluate performance after learning 100 objects (10 of each MNIST class), and in the context of noise. In their primary simulation, noise was applied at learning in the form of pixel blurring, rather than at inference time (although they also explored conditions such as occlusion, which we do not consider here). We were concerned that any noise robustness in GridCellNet might in fact be attributable to the pre-processing stage, and so we applied feature-level noise in the form of randomly flipped bits (changing a 0 to 1 or vice-versa) at inference. For each feature vector in the 5×5 or 4×4 visual environment, n randomly chosen bits are flipped from their original values (Figure 2.12a). We demonstrate the ability of GridCellNet to correctly recall the precise object that it

observes as a function of the amount of noise.

We begin with results from the RLF setting. Figure 2.12b shows that in our main condition with a higher dendritic threshold ($\theta^{\text{loc}} = 20$), GridCellNet remains robust up to around 30-35 flipped bits, or around 1/4 of all bits. This is also slightly more than the number of on-bits in the feature vector without noise (29). Such robustness is consistent with the known advantages of sparse representations for robustness [Ahmad and Scheinkman, 2019]. We also show that, using the same dendritic threshold that was used for *classification* with 20 training objects per class ($\theta^{\text{loc}} = 16$), recall is generally not as high-performing.

This drop in performance is especially evident in the low noise setting. Interestingly, there is actually an improvement in recall accuracy as a small amount of noise is added, particularly for this lower dendritic threshold. This appears to be a result of noiseless feature vectors often being compatible with multiple learned objects when the matching threshold is easily surpassed. In this case, the network never settles on any one particular example that is close to the input. The introduction of noise can eliminate spurious matches to learned representations, as the features that truly relate to the learned representation are more robust to noise (i.e. more bits can be flipped before they will fail to exceed the matching threshold). Importantly, in either the low or high threshold case, there is a regime in which recall accuracy approaches 100%. In our Discussion (Section 2.6), we will return to the significance of different dendritic thresholds and improvements with noise.

We replicate these results in the ILF setting (Figure 2.12c). Interestingly, the network appears to be able to better handle the noiseless condition, regardless of the dendritic threshold. The feature extraction method employed in the RLF setting likely encourages more invariant representations, which are therefore less able to support distinguishing different members of a common class. In other words, representations that support object classification (where RLF features were superior) are less useful for recalling a precise object instance.

It is challenging to make direct comparisons to related work due to differences in approaches. For example, Bicanski and Burgess [2019] manually selected 9 salient feature locations for each object, rather than sampling the entire input space as we do. Despite

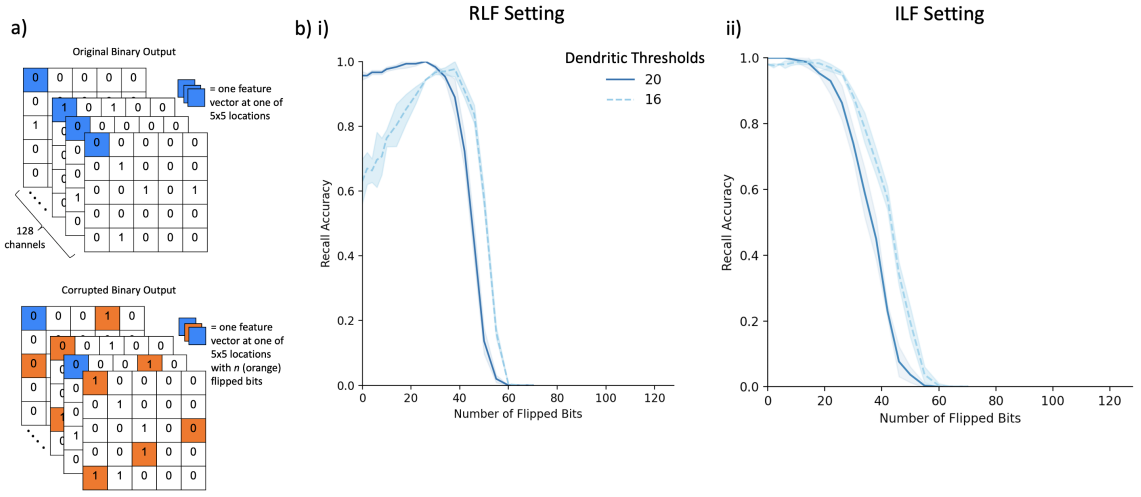


Figure 2.12: *Recall of Training Examples as a Function of Noise.* Following previous work, the ability of the system to recall objects learned during training is evaluated. a) To determine the robustness of this process to corrupted inputs, noise is applied at the feature level in terms of flipped bits. n random bits are flipped in *each* of the 5×5 (or 4×4) input features individually. b) Accuracy of recall shown as a function of flipped bits, with two different dendritic threshold values (θ^{loc}). Results are shown for the RLF (i) and ILF (ii) settings. The shaded region shows the 95% confidence interval of the mean across three random seeds.

this, it is worth highlighting that we observe successful recall in spite of remarkable inter-stimulus similarity. In particular, many related studies showing successful recall used stimuli with often high inter and intra-class diversity, such as objects and scenes as varied as teddy bears vs flower pots [Hoang et al., 2020], or a Greek statue vs an image of a brain [Bicanski and Burgess, 2019]. Such variety significantly simplifies the task of recalling a particular learned object, whereas any particular MNIST digit can be very similar to another within its own class, or even to that of another class.

Finally, while the total number of *possible* sensations (25 or 16) is much higher than in Bicanski and Burgess [2019] (9 features), we highlight that recall normally occurs long before all of these features are sampled. The efficiency of recall in the absence of bit-wise noise is shown in Figure 2.13.

Specifically in the RLF setting, the mean number of sensations required for recall is 4.2 ± 0.1 (mean model performance across three random seeds and 95% confidence interval) when there are 0 flipped bits and $\theta^{\text{loc}} = 20$. In a noisy regime, such as 20 flipped bits, we actually observe a slight decrease in the mean number of sensations, to 2.8 ± 0.1, likely

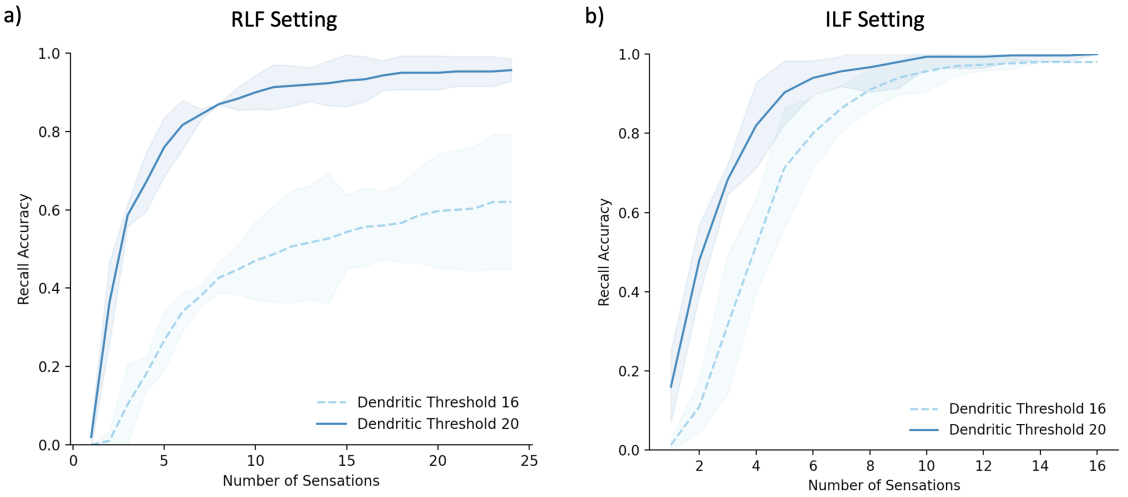


Figure 2.13: *The Recall Accuracy as a Function of the Number of Sensations.* We show the total recall accuracy across progressive sensations; note that in these results, no bit-wise noise was included. The shaded region shows the 95% confidence interval of the mean across three random seeds.

for the same reason that noise can be beneficial given a lower dendritic threshold. Thus, we observe that GridCellNet operates both robustly, as well as efficiently, in the recall regime.

2.5.2 Predictive Representations

In Section 2.4.1, we demonstrated that GridCellNet can often perform classification before the entire input sequence is experienced. Similarly, Section 2.5.1.1 demonstrated that, given a previously seen object example, the internal representation can rapidly converge to the same representation experienced during learning. In both of these cases however, the utility of these rapidly developed representations to predict and visualize unsensed regions of the input was not explored. At any given time however, GridCellNet predicts the sensory input it will receive after a movement. These predictions are based on the currently active representations, which can be viewed as GridCellNet's working hypotheses about the object being sensed.

In this section, we will visualize these representations in both the settings of novel objects (i.e. image reconstruction), and familiar objects seen during training (i.e. associative memories). We highlight that (like humans) GridCellNet makes predictions at the feature,

not pixel-level [Rayner and Pollatsek, 1983, Hayhoe et al., 1991, Holm et al., 2008]. Fortunately, the use of a decoder neural network will enable us to visualize these abstract features directly.

2.5.2.1 Image Reconstruction

We begin with the setting where we wish to visualize GridCellNet's predictions of unsensed parts of the image, given partial inputs of an object it has never seen before. The question of interest is whether the recovered image will i) look like a plausible example of that object class, and ii) bear some resemblance to the particular novel example presented.

To visualise GridCellNet's representations during inference, we extract the sequence of sensed and predicted features at each progressive movement. As the sensor passes over the sequence of inputs, we accumulate the ground-truth sensations previously experienced, as well as the prediction for the next movement. Importantly however, once the system has converged to the representation of a single object, all future accumulated representations are based solely on predictions from the inputs received up until the time of convergence.

We begin with results from the RLF setting. Figure 2.14 shows examples of GridCellNet restoring from memory examples that closely match the input. In the early stages of inference, the internal representation consists of a small number of previously perceived features, and the next predicted feature. While GridCellNet's prediction is only queried for a single location (centred at the yellow highlighted patch), the decoder is fed the full representation array, and attempts to reconstruct the entire input. Furthermore, any given feature vector is an abstract representation of a 16×16 pixel region (25% of the entire input space). As such, un-queried regions are not completely empty. At the point at which single-object convergence occurs, enough features have been sensed that the decoded image often appears recognizable to a human. After then evaluating predictions at every unseen location, we observe that GridCellNet can recall an example from memory that is similar to the input, thereby enabling reconstruction.

Note that although there will be overlap in the receptive fields of the different sensed inputs, reconstructing a detailed representation of the object requires GridCellNet to predict the abstract representation at a particular foveation, given only the abstract representations

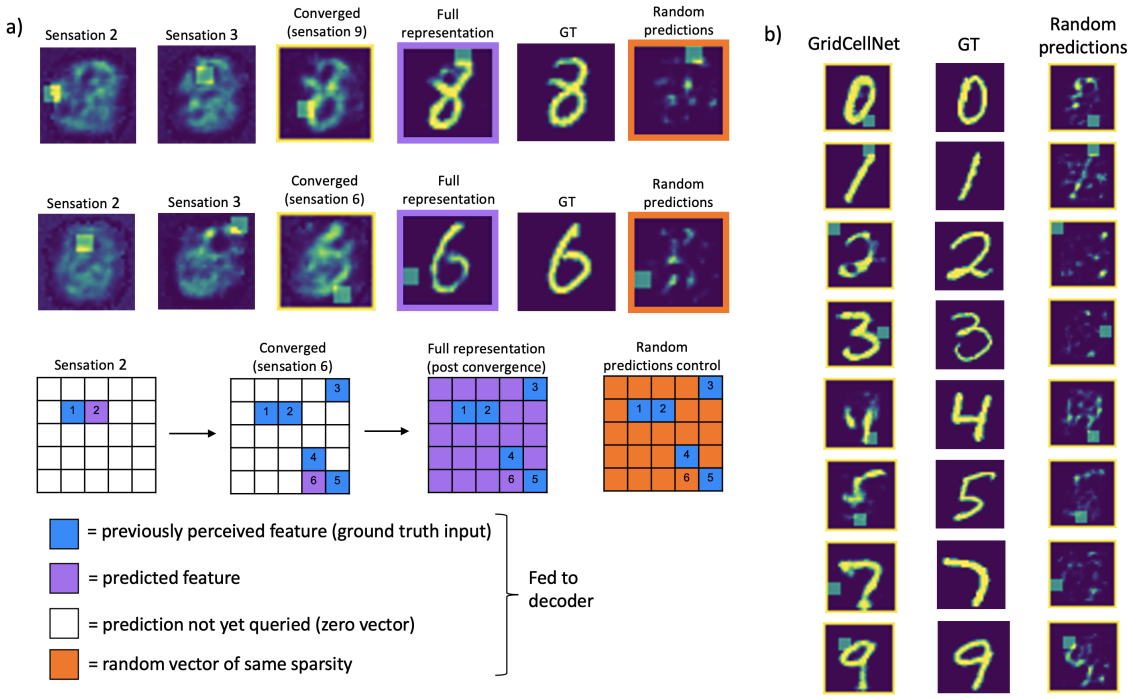


Figure 2.14: *Predicting Unsensed Parts of Novel Objects (RLF* 5×5 *Visual Environment).* a) We use the sensed and predicted features of GridCellNet and a decoder network to visualise the system's representations. We show two example prediction sequences (top). GridCellNet predicts its next sensation (centred at the yellow highlighted patch) based on prior input. After converging to a single representation, we visualize the system's predictions of the entire image (purple border). A diagram of the correspondence between past sensations and upcoming predictions is given at the bottom, matching the representations seen for the '6' in the middle row. At each sequence step, the entire 5×5 grid of feature representations (each a vector of length 128) is fed to a separately trained decoder. We also show a random predictions control (orange border), where predictions are replaced by random binary vectors of the same sparsity. Note "Sensation 1" is not shown, as at this time, there is no prediction for the network to make. b) Several randomly-selected examples from the other classes. Results are shown from GridCellNet when trained on 5 examples per class, and $\theta^{\text{loc}} = 12$. GT = ground truth.

at other foveations, i.e. without direct access to the representation at a pixel-level. Despite this, one might wonder if there is sufficient data in the initial ground-truth sensations before convergence to accurately reconstruct the input. In particular, it might be possible to pad out the unsensed, predicted regions with random vectors of the same sparsity as the ground-truth features, and yet still achieve accurate reconstruction. To rule out this possibility, we provide a random predictions control, where the ground-truth features up until convergence are still provided, but predicted representations are replaced with such random vectors. These are included for the two demonstrative digits in Figure 2.14a, as well as for an array of additional digits in Figure 2.14b. As can be seen, these control reconstructions are almost entirely meaningless, and so GridCellNet's predictive functions are crucial to the plausible reconstructions observed. While some examples of GridCellNet's representations are far from perfect (in particular the '2' and '4'), the results provide evidence that these feature-level predictions could support additional downstream tasks other than classification.

We note that we limit visualization to situations where the network converges to a single representation (e.g. a single example of a '9', as opposed to several compatible '9's), as the *decoder* in its current form is unable to make sense of the input if the representation at inference includes a union of object representations. Such a union of representations will result in a vector with a different sparsity to that which the decoder has been trained on. This is not to say that GridCellNet's predictive representations before this point are not also meaningful, they are simply more difficult to visualize without bespoke training of the decoder. When GridCellNet is trained on 5 examples per class (the current setting), single-object convergence occurs for around 25% of all objects. On a small subset of these (<5% of the single convergence objects, or around 1% of all objects), GridCellNet fails to predict a feature for every region of the image. Note that this is not the same regime as in Section 2.5.1 or below where we consider associative memories.

We also assess image reconstruction in the ILF setting, where receptive fields are smaller (7×7 pixels) and non-overlapping. Figure 2.15 shows the results from such an evaluation. Aside from the change in the visual environment, the key difference is that the decoder is trained and evaluated using restricted patches rather than the entire visual space. We once

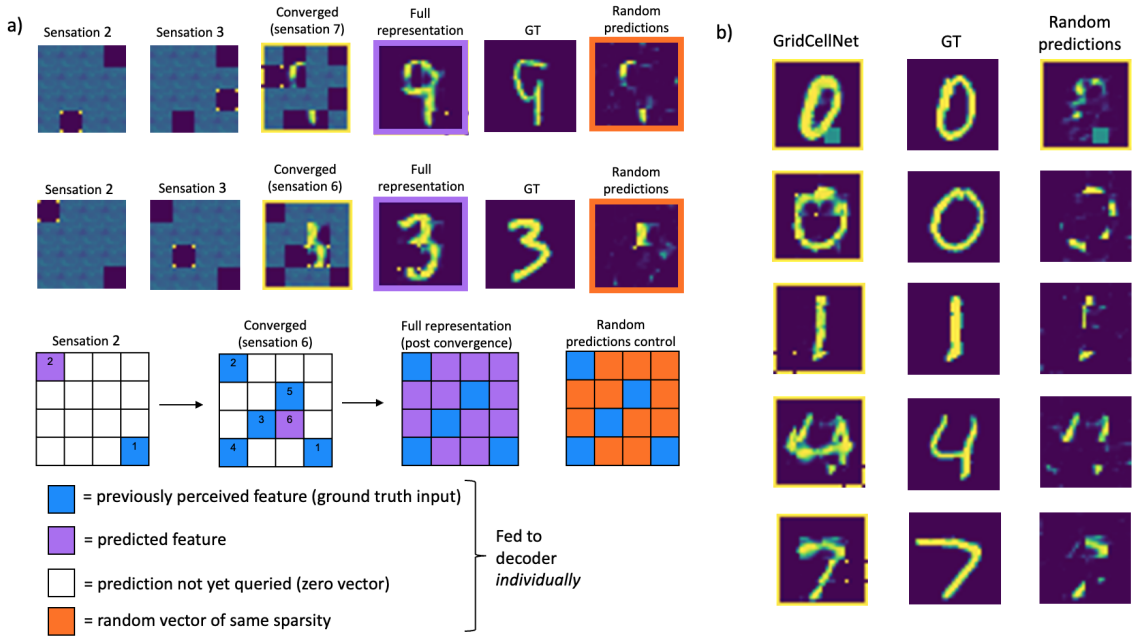


Figure 2.15: *Predicting Unsensed Parts of Novel Objects (ILF 4×4 Visual Environment).* a) As in Figure 2.14, we show two example prediction sequences (top). GridCellNet predicts its next sensation (centred between four highlighted points) based on prior input. After converging to a single representation, we visualize the system's predictions of the entire image (purple border). A diagram of the correspondence between past sensations and upcoming predictions is given at the bottom, matching the representations seen for the '3' in the middle row. At each sequence step, the next feature vector (dimension 128) is fed to a separately trained decoder. We also show a random predictions control (orange border), where predictions are replaced by random binary vectors of the same sparsity. Note "Sensation 1" is not shown, as at this time, there is no prediction for the network to make. b) Several examples from the other classes. Results are shown from GridCellNet when trained on 20 examples per class, and $\theta^{\text{loc}} = 16$. GT = ground truth.

again observe that GridCellNet is able to reconstruct plausible representations of novel objects based on its stored representations.

2.5.2.2 Associative Memories

In the above results, we explored predictive representations when GridCellNet processes novel objects. We now consider the setting of predictive representations when the image being viewed is a familiar stimulus that was seen at training time. The recovery of the sensory input can therefore be considered as an associative memory.

We use the same process as in the setting of image reconstruction, albeit here GridCellNet is trained on 10 examples per class (100 objects total), and with $\theta^{\text{loc}} = 20$. Unlike the

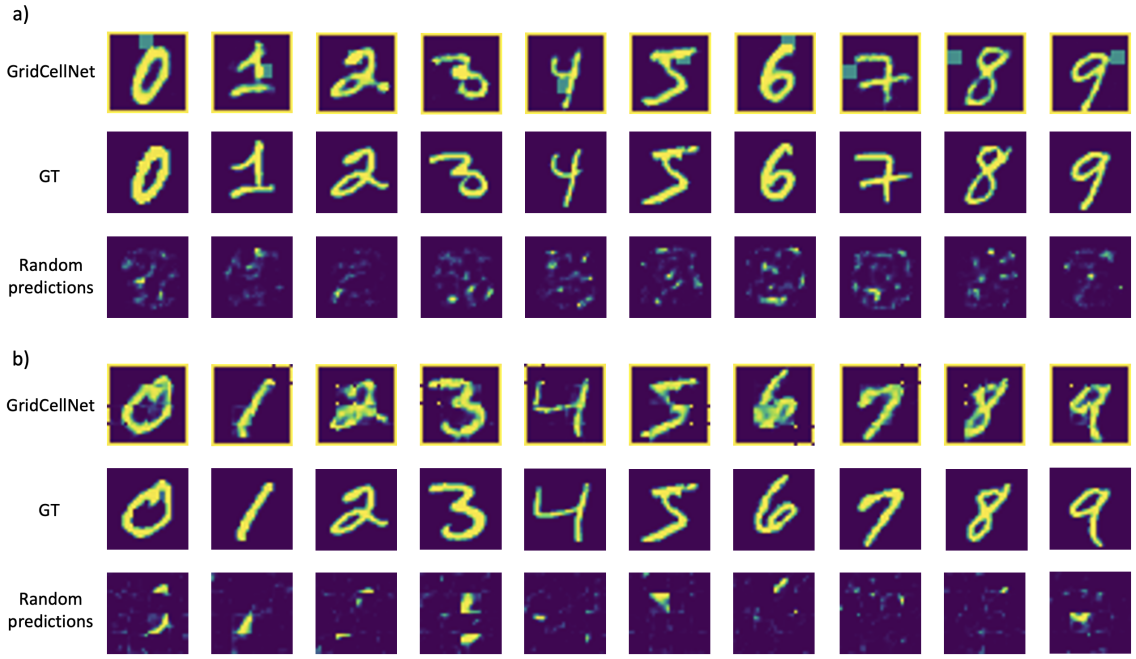


Figure 2.16: *Predicting Unsensed Parts of Familiar Objects.* Random examples of recovered objects from the RLF (a) and ILF (b) settings. The top row shows the reconstruction from GridCellNet's representation, and the bottom row a control where all predictions are replaced with random bit-vectors of the same sparsity. GT = ground truth.

previous results, where it is sufficient for the predicted image to resemble a plausible member of the class of interest, we would like the recovered image to closely match the actual query image, in spite of potentially other memorized images that bear a close resemblance.

We provide random samples from both the RLF (Figure 2.16a) and ILF (Figure 2.16b) settings. As can be seen, the images recovered closely match the original stimuli. As such, with a common architecture, GridCellNet is able to predict unsensed regions of both novel stimuli, as well as previously perceived stimuli.

2.5.3 Continual Learning

GridCellNet is not designed to specifically address the challenging issue of continual learning. Due to its use of Hebbian learning rules and sparsity however, it benefits from natural robustness in this setting, which is here explored in comparison to an LSTM trained with back-propagation of error.

Continual learning is evaluated in two main learning splits (summarised in Figure 2.17a), and in the context of the RLF setting. The classifiers are either trained on the first 5 MNIST digits and then the next 5 (i), or the first 9 and then the final digit (ii). Accuracy is always evaluated across all classes, and so the possible performance of an ideal classifier is also shown. Note that this is different from the 'split-MNIST' evaluation sometimes used for continual learning [Zenke et al., 2017], as it requires an optimal classifier to accumulate information to eventually be able to recognise any of the classes seen over the course of learning. The classifier heads are not modified beyond standard learning rules. Additionally, weights are never artificially fixed.

Before examining the results, it is worth clarifying how learning proceeds in GridCellNet. For a back-propagation trained network to perform well, training examples are generally interleaved and shuffled for each epoch. In GridCellNet, there is no significance for using an interleaved vs. a blocked training regime. As such, the default form of learning is carried out in a continuous fashion, each class following the other and with only a single pass over each object, with no returning to previously learned objects or requirement for multiple classes to be observed in an alternating fashion. As a result, GridCellNet is in fact naturally capable of a continual learning regime that would be considered even more challenging than the 5-5 or 9-1 split we simulate (Figure 2.17a.iii). While we show results for a 5-5 and 9-1 split, we do not show the results of such a '1-1-...-1' split. In particular, an LSTM classifier would be essentially bounded to 10% accuracy, classifying everything as the most recently learned class. Despite such a setting being obviously catastrophic for the LSTM, it is worth noting that this is a plausible situation for an agent in the natural world to encounter. Imagine, for example, discovering a cluster of trees filled with a new kind of edible fruit, with no other fruit varieties nearby; all learning must then proceed on that single class. Such concentrations are common in the natural world [Condit, 2000].

The results in Figure 2.17b) demonstrate that GridCellNet is perfectly suited to gradually accumulate new types of objects that it encounters, with the accuracy across all the possible classes consistently improving as new classes are introduced in learning. In contrast, the LSTM demonstrates the well-known property of catastrophic forgetting. As soon as a new task is presented, knowledge required to solve the new classification problem rapidly

over-writes that used for previous tasks. The results also reinforce how GridCellNet's use of a single pass over each object (rather than multiple epochs) enables very rapid learning. Note, in particular, the essentially instantaneous jump in performance of GridCellNet; unlike the LSTM, GridCellNet does not need to repeatedly re-visit examples it has seen before in order to learn.

Note that these results are in the context of non-arbitrary (i.e. fixed) input sequences, as well as 20-training examples per class. Thus we compare the performance of the systems in the setting where the LSTM achieves its highest performance, and stands a chance of performing better than GridCellNet, even though this is not a particularly naturalistic setting (i.e. requiring more than a dozen examples per class, and spatiotemporally fixed sensory input sequences). Despite this advantage (note in particular the strong performance of the LSTM when trained on the first 9 classes in the 9-1 split), the LSTM is in a sense a victim of its own strength, and rapidly fails when attempting to extend its learning to a novel class. If we were to evaluate performance in a more natural setting with arbitrary sequences and e.g. 5-shot learning, GridCellNet's advantage would be even more stark. Finally, we note that our evaluation is not a strict continual learning setting, in that the RLF pre-processing was performed across all 10 classes from a different sub-section of the data-set, although this benefit is shared by both GridCellNet and the LSTM.

2.6 Discussion

We have presented a novel approach to the challenge of visual object recognition given an arbitrary sequence of feature inputs sampled across space, a form of out-of-distribution generalization. Robustness to novel sequence trajectories, as well as meaningless objects, is achieved through the use of grid cells to model the location of features in the reference frame of an object. This robust classification is often rapid, occurring after only a fraction of the total input space is sampled, and can occur alongside the recall of familiar objects. Finally, GridCellNet takes advantage of rapid Hebbian-style weight updates and sparsity to enable few-shot and continual learning robustness, and predictive components to enable the completion of partially sensed images. This combination of challenging requirements is captured by the SCIFFCAR task settings. Future versions of the proposed architecture

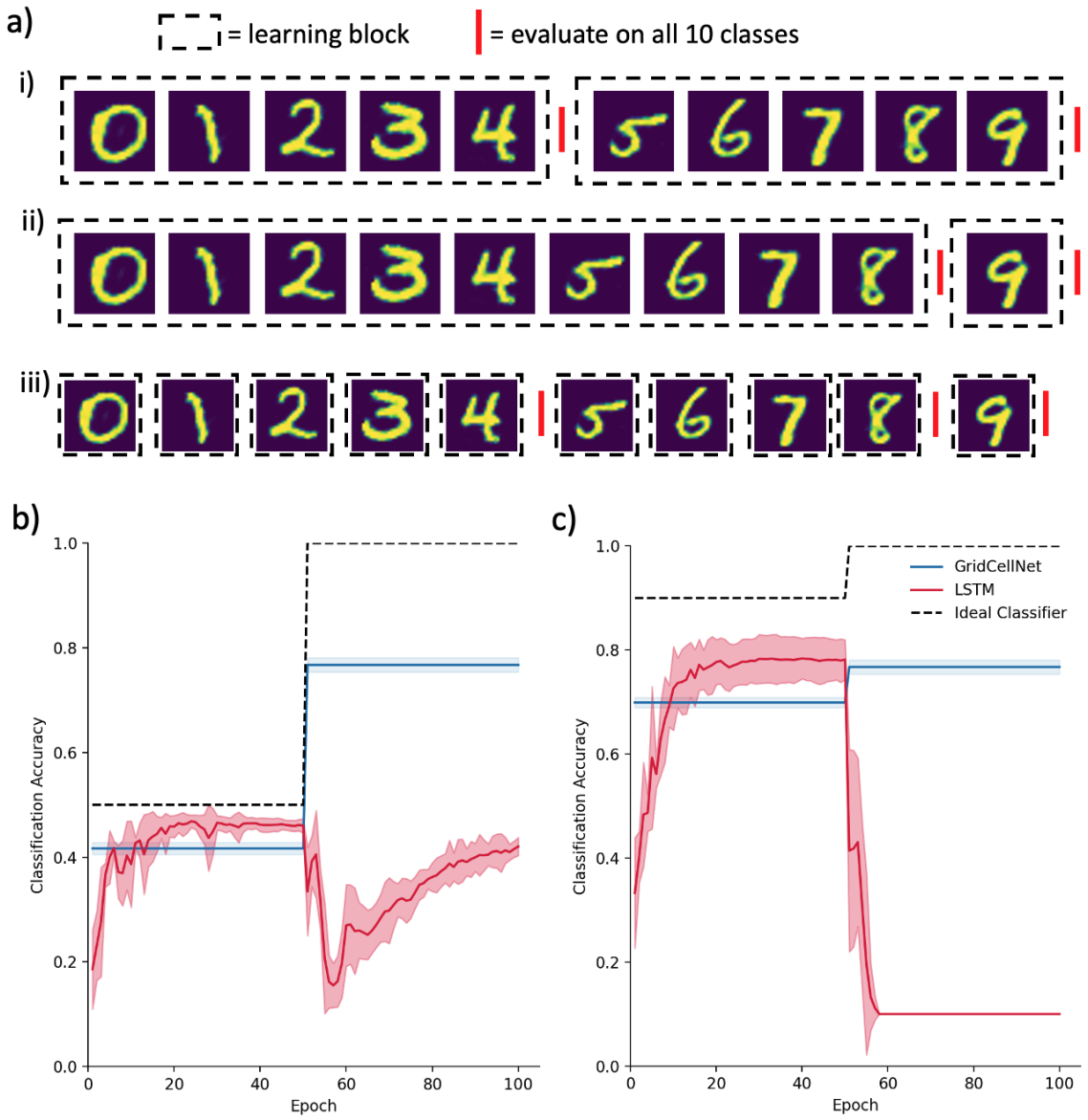


Figure 2.17: *Robustness in a Continual Learning Setting.* a) The training and evaluation splits used for the LSTM are shown in (i) and (ii), and for GridCellNet in (iii). For the LSTM, training examples are interleaved within a particular learning block. b) Results of the 5-5 split (left) and 9-1 split (right). As evaluation accuracy is always assessed across all 10 classes (even before they have been observed during training), the hypothetical performance of an ideal classifier is also included. The shaded regions show the 95% confidence interval of the mean across three random seeds.

that enable greater integration of back-propagation and rapid Hebbian learning will likely be important to realising its full potential.

While this work has focused on the machine learning implications of such an approach to visual processing, it is worth considering the potential value of GridCellNet as a model of primate vision. Our work addresses limitations associated with related models, such as the inability to handle natural images [Lewis et al., 2019], the absence of a fully internal reference frame [Bicanski and Burgess, 2019], and a lack of generalization to novel objects [Lewis et al., 2019, Bicanski and Burgess, 2019]. GridCellNet resolves these limitations while remaining consistent with the known neuroanatomy of the primate visual system [Suzuki, 1996, Buckley et al., 2001, Barense et al., 2005] and biologically established learning mechanisms. Furthermore, we find that GridCellNet processes spatially distorted objects in a broadly similar manner to humans [Holm et al., 2008, Orlov and Zohary, 2018]. We believe that this work therefore supports the notion that the brain may use grid cell computations when performing visual object recognition, and that this might underlie some of the visual tasks in which humans still outperform engineered systems. Further discussion of our results and their context is provided below.

2.6.1 GridCellNet and Binding

Learning in GridCellNet is a form of rapid binding between arbitrary sensory features and their location in the internal reference frame of an object. The presented model therefore represents a reasonable approach to token creation [Treisman, 1999] using grid cells [Hawkins et al., 2019, Whittington et al., 2020], albeit here demonstrated in visual space [Bicanski and Burgess, 2019]. Our primary task demonstrated that these tokens could subsequently be leveraged for object recognition (i.e. conjunction detection) that is sensitive to the spatial arrangement of features, while being potentially invariant to the location of objects in an external environment. The ability to rapidly form associative representations and use these for a variety of downstream tasks, such as predicting unsensed (e.g. occluded) features, supports the more general utility of such tokens. It is worth emphasizing the particular strength of grid-cell path integration in this context, which allows the map used for binding features to be traversed using pathways that have never been experi-

enced. Naturally, such binding with grid cells would be complementary to other forms of visual feature binding that make use of hard-coded associations, and which are more appropriate for parallel processing [Riesenhuber and Poggio, 1999a, Hommel and Colzato, 2009, VanRullen, 2009, Eguchi et al., 2018]. We further note that previous work has proposed that grid cells might serve as a basis for arbitrary *hierarchical* binding [Hawkins et al., 2019, Whittington et al., 2018, 2020], although work in this area remains limited, and simultaneously processing multiple levels of feature abstraction is not something we have considered in our model. We thus highlight this only to note the potential relevance of architectures such as GridCellNet for more complex binding representations.

2.6.2 Grid Cells for Visual Object Recognition

This work was partly motivated by the observation that humans solve our opening task naturally when performing saccades, and that grid cells might enable a biologically plausible solution. It has been proposed that humans might perform object recognition in a variety of sensory modalities by making use of grid cell computations [Hawkins et al., 2019], including in vision [Bicanski and Burgess, 2019]. As we note in our results, the ability to achieve the same classification performance on an arbitrary sequence input represents a form of out-of-distribution generalization, as the probability of a particular sequence trajectory reoccurring is astronomically low. This leaves a classifier with the option of either ignoring spatial relations entirely, or implementing a method to handle arbitrary sequences. Our work demonstrates that grid-cell computations represent a credible approach to the latter when implemented in a machine learning model.

In Bicanski and Burgess [2019], the authors used features extracted from multiple locations of an image to perform a task with some similarities to our own, and this important work was the first biologically plausible model to demonstrate such capabilities. As in the work presented here, the features were sequentially fed to a classifier that integrated these into a learned representation. When subsequently challenged to recall which of a handful of memorized images were presented, the system successfully did so, even under settings such as partial occlusion. Importantly however, their focus was on visual recognition *memory*, and images used during training were the same as those used at

evaluation time. There was therefore no need for the system to generalize to unseen examples, recognising the commonality between different instances of a class. Consistent with this, the system's working hypothesis was always constrained to a single object. Our system is designed to represent multiple compatible hypotheses at any given time point. These multiple representations then serve as features for GridCellNet's final classification decision, supporting generalization. Our work therefore represents the first demonstration that grid cell-like computations can be leveraged to enable generalization on a visual task to unseen examples of an object class.

2.6.3 Empowering Agents with Flexibility

Providing a system with the flexibility to perform well with novel sampling sequences has obvious appeal. Together with performing classification without traversing the entire sequence space, an agent employing such object recognition could sample the most informative regions in a principled manner, and thereby operate much more efficiently. This can be contrasted to an agent that is constrained to sample every point, always following the same sequence, such as a raster scan across the image. An interesting area for future investigation would be pairing GridCellNet with a reinforcement learning agent that can learn to optimally control the movement of its sensor given the system's working hypothesis.

2.6.4 Translation Invariance

An issue relevant to both neuroscience and machine learning is that of translation invariance as it applies to classifiers that rely on sequential inputs. GridCellNet uses an internal reference frame to encode the spatial relations of features and perform classification. As such, it is agnostic to the location of an object in the external environment, and represents a step towards classifiers capable of such translation invariance.

Bicanski and Burgess [2019] argued that translation invariance would be possible in their sequential, grid-cell based model following additional changes. As the model was presented, however, it assumed that all of the objects were aligned in the same grid cell reference frame across training and testing. Such an assumption requires perfect path

integration, and for objects in the environment to always be correctly aligned with the grid cells' responses. As a method for the grid cells to re-align based on sensory inputs was not presented, translation invariance does not automatically follow. While we do not simulate noisy path integration, GridCellNet uses sensory inputs and iterative refinements with self-movement to infer positions in an internal reference frame, addressing these requirements. Research in the robotics literature that used an internal reference frame to achieve translation invariance [Pezzementi et al., 2011] represented an important proof of principle, but this work was limited to a tactile (as opposed to a visual) task, and it did not explore generalization to novel instances of a class. GridCellNet addresses these limitations, serving as an important step towards translation invariance for sequential classifiers.

An important limitation in our approach is that translation invariance in GridCellNet would require that the pre-processing/foveal-response map we generate is translation equivariant following sufficiently large, discrete movements, and invariant to small movements. In the RLF setting, the pre-processing stage also made use of a parallel, rather than sequential process. Thus, in order to deploy GridCellNet to natural translated images, the same pre-processing we used for our patch-based auto-encoder (ILF setting) would need to be used. However, the robustness of these feature representations (e.g. to small image translations) would also need to be improved, and the process would likely benefit from a less discrete view of visual space. Exploring architectures and training regimes that can support this represents an interesting area for future work.

2.6.5 Recall

While classifying objects is a useful ability, so too can be recalling a particular learned instance of an item. Humans appear to be quite capable at this task for a diverse range of objects [Brady et al., 2008]. Such recall has also been the primary evaluation in previous systems relying on path-integration supported recognition [Pezzementi et al., 2011, Browatzki et al., 2014, Lewis et al., 2019, Bicanski and Burgess, 2019, Hoang et al., 2020]. As such, we also demonstrated GridCellNet's ability to perform this task, including showing that recall was robust even at considerable levels of noise at the feature-level. This finding is consistent with GridCellNet's use of high-dimensional, sparse represen-

tations, which can be robust to noise [Ahmad and Scheinkman, 2019]. Unlike other biologically plausible models [Bicanski and Burgess, 2019], GridCellNet relies entirely on an internal reference frame to perform this recall, eliminating an important issue of neuro-plausibility, and satisfying a requirement for translation invariance as discussed separately.

An additional benefit of GridCellNet is that recall operates at the same time as the computations that support recognition of a general object category. While we observed more robust memory recall in the RLF setting when using a higher dendritic threshold than that used for classification, the common value of $\theta^{\text{loc}} = 16$ appeared compatible with some degree of both recall and classification.

On the one hand, it is possible that the brain might dynamically alter the sensitivity of the dendritic threshold to suit various purposes. In particular, a lower threshold supports classification, but can be too accommodating when attempting to recall a specific example. The possibility of neurons implementing such dynamic threshold adjustments is supported by both detailed computational models and experimental recordings of neurons [Jadi et al., 2012].

Alternatively, we observed that the addition of small amounts of noise could actually improve recall, likely by ensuring the system did not spuriously believe a partly matching feature was similar to a learned representation. Unlike such spurious matches, the true feature that aligns with the learned representation should be more robust to this added noise, and as such recall can successfully converge to the correct learned object. This benefit of noise is interesting as we observed it rescue recall accuracy to nearly 100% in the setting of a fixed dendritic threshold that could also be used for classification. The implication is that even if a neuron cannot rapidly alter its threshold for dendritic spikes, intentionally adding noise to the feature-level representations could be used to dynamically switch the system from a classification state to a memory recall state.

2.6.6 Predictive Representations

One appealing aspect of GridCellNet is its predictive nature. While this is crucial to how it performs inference, we demonstrated that this has the additional advantage that

the system can predict unsensed regions of the input. These predictions take place even before inference, but for the sake of being able to visualize these directly, we demonstrated examples of these predictions when GridCellNet's representation had converged to that of a single object.

As we noted in our results, GridCellNet makes its predictions at the feature level, rather than at a pixel level. The separately trained decoder then enabled us to visualize these feature-level predictions as images. This is significant, as experimental evidence suggests humans can predict visual representations across saccades at an abstract feature level [Hayhoe et al., 1991, Holm et al., 2008], but not at a pixel-like level [Rayner and Pollatsek, 1983]. As such, GridCellNet's capabilities align well with the psychological literature on such abilities in humans.

2.6.7 Continual, Few-shot, and Rapid Learning

We demonstrated that the proposed architecture naturally displays several desirable properties. As a consequence of the use of Hebbian learning rules and sparse representations, GridCellNet is robust in a continual learning setting, enabling it to learn to recognise a novel object without obliterating its previous classification abilities. Notably, this can take place even when the novel class is shown in isolation, as demonstrated in our 9-1 split. In other words, learning does not require that multiple new classes are provided in a batch. This is appealing for its realism, as it seems odd to assume that in the real world one would always encounter novel objects in batches with other novel objects, rather than in isolation.

All of our evaluations were also conducted in the setting of few-shot learning, where GridCellNet excels. We emphasize that while the accuracy achieved by GridCellNet is not as high as some other approaches to few-shot learning [Wong and Yuille, 2015, Lake et al., 2015], our intention is not to propose the model as a strong solution to that general setting. Rather our purpose is to show that in the context of few-shot learning, the use of grid cell representations can provide robustness to unpredictable input sequences, which should have downstream benefits for embodied agents.

Finally, the weight updates used also support extremely rapid learning in GridCellNet

from an algorithmic perspective - given a novel object, the system need only perform a single pass and set of weight updates. Unlike the multiple epochs typical of back-propagation of error, GridCellNet does not need to repeatedly revisit a given example to perform learning. This is thus consistent with the ability of humans to learn a novel object given distinct parts, and within a short period of time, recognise that same object [Króliczak et al., 2003, Orlov and Zohary, 2018]. Similarly, during inference, GridCellNet's internal state updates require only a single set of steps for each saccade/feature input; there is no need for multiple iterations to take place in order for the internal state to converge for each feature presentation. This is in keeping with the finding that saccades in primates typically last <300ms [Salthouse and Ellis, 1980], which represents a challenging constraint for recurrent algorithms that require dozens of iterations for any given saccade.

2.6.8 Alternative Architectures for Sequential Inputs

Transformer Networks While we compare our architecture to an LSTM, it is possible that transformer networks [Vaswani et al., 2017] would perform better in this setting. They currently represent the state of the art in many sequence-based tasks, including visual tasks [Dosovitskiy et al., 2021], and explicitly encode positional information. However, this positional information is usually absolute and derived from an external reference frame, such as the dimensions of an unrolled image. This brings with it the issues for translation invariance discussed previously, although more recent work has explored using relative positional information in order to address this. Interestingly, it has been empirically found that transformer networks using relative positional information perform less well on classification tasks [Wang et al., 2021].

More importantly, even when relative positional information is used, the embeddings explored in the literature are one-dimensional [Shaw et al., 2018], which in image space corresponds to displacement in a flattened vector [Dosovitskiy et al., 2021]. Such an approach however assumes a two-dimensional space with fixed dimensions. In particular, converting displacements in two dimensions to one dimension is only possible if the lateral displacement for any given row is fixed (e.g. an object always takes up a grid of 4×4

fixations). This approach is even more problematic if the fixations corresponding to an object have a non-square structure. Grid cells handle this situation naturally by modelling 2D space with 2D coordinates. Interestingly, very recent work [Whittington et al., 2022] has shown that, like RNNs, transformer networks with sufficiently flexible positional encodings and training can *learn* to develop grid-cell like positional encodings. Thus, as with RNNs, we do not claim that a sufficiently well-designed and pre-trained transformer network would not eventually be able to solve this task in a similar manner to GridCellNet. Rather, we set out to show the benefit of the inductive bias of grid-cell computations in a machine learning architecture for learning rapidly.

It is also worth noting that, while not explored here, grid cells can in principle encode 3D structure [Klukas et al., 2020]. Transformer networks already suffer from efficiency issues with long sequences, owing to the massively parallel (and likely biologically implausible) self-attention computation that they perform. The introduction of a third dimension in the input representation would be problematic for the efficiency of this parallel computation. The performance of transformer networks on our task and the generalization of GridCellNet to 3D objects would therefore be worthwhile topics for future investigations.

Graph Neural Networks and Predictive Coding Architectures Graph neural networks [Scarselli et al., 2009] may also be a viable basis for the tasks we outline here. However, we are not aware of work in graph-neural networks where inputs are constrained to be sequential, while implementing an internal reference frame for visual space together with Hebbian-learning based object recognition. It is likely that such elements will be important to achieve the same performance we see across the variety of tasks explored (SCIFFCAR).

Recent work by Salvatori et al. [2022] demonstrated an interesting approach where the use of local predictive coding learning rules (as opposed to standard back-propagation of error), enabled the design of architectures with arbitrary graph structure. Like our approach, this model was not specifically designed for a single task, but instead demonstrated strong performance across a variety of tasks, including classification, associative memory, and image reconstruction, and including on complex data-sets that we do not explore here. We view these approaches as complementary, as the work by Salvatori et al.

[2022] was focused on the setting of parallel (e.g. retinotopically structured) inputs, such that query features were associated with an established spatial location, rather than needing to be inferred as in our sequential-input task. This is thus a separate problem setting to the one we consider, and would be relevant when processing retinotopically structured inputs, such as predicting occluded features given a single glimpse of an object or scene. Similarly, [Salvatori et al., 2021] explored biologically plausible associative memories using predictive coding, where once again the setting was one of a parallel input where a sub-portion (potentially as little as 1/16th of the image), was presented in a specific region of the input space in order to trigger robust recall. Given this difference in the problem setting, these represent complementary approaches to our work.

2.6.9 The Strengths of Deep Learning

It is worth noting that while our results demonstrate multiple benefits of the proposed GridCellNet architecture, they also reinforce some of the strengths of neural networks that use continuous activation values and learn via back-propagation of error. In particular, while the LSTM does not have the inductive bias of GridCellNet to perform path integration in a few-shot setting, it makes rich use of the features available, and scales well as more learned examples are encountered. In regards to the former, the results from our false-motor information ablation demonstrate that GridCellNet does not extract as much information as is available from the features for predicting the target class. In terms of performance with more training data, the ceiling for accuracy appears to be higher for the LSTM as the number of learning examples grows, and indeed our own preliminary results suggest that with thousands of training examples, GridCellNet's memorized representations begin to interfere with one another. It is likely that the learning and inference mechanisms proposed for GridCellNet could work well in tandem with more standard deep-learning approaches, where these might be implemented via more biologically plausible mechanisms, such as those leveraged in Salvatori et al. [2021, 2022].

2.6.10 Additional Data-Sets

The results presented here are based on a challenging formulation of MNIST [Lecun et al., 1998], which we summarised with the acronym SCIFFCAR, and where, amongst other limitations, local feature representations are sequentially fed to a classifier in a few-shot learning setting. We acknowledge that demonstrating the architecture on additional data-sets would be valuable. We would like to highlight however that MNIST is not trivial in the setting we explore. This is exemplified by the results of the baseline models, as well as the fact that previous, related models to our own have been restricted to synthetic or non-generalization settings. This is also consistent with other domains in object recognition where MNIST remains sufficiently difficult so as to be unsolved, including adversarial examples [Schott et al., 2019, Tramèr et al., 2020a] and robustness to general noise corruptions and out of distribution translations [Mu and Gilmer, 2019]. Finally, we once again highlight that under the combined setting we explore, even humans can find object recognition relatively challenging [Króliczak et al., 2003, Holm et al., 2008, Orlov and Zohary, 2018].

2.6.11 Conclusion

In this chapter, we have explored the benefit of using grid cells as a basis for rapidly binding feature representations to an internal object reference frame. The strength of such an approach was explored on an object recognition task, demonstrating superior robustness over traditional machine learning methods. In addition, secondary benefits of the architecture for a variety of visual tasks were explored. Finally, we discussed the consistency of these results with current knowledge of the primate brain and human perception. In our over-arching thesis discussion, we will address how the computations we have explored here could operate in a complementary manner with the forms of binding we examine in the next three chapters.

3 | Delay-Lines and Robust Translation Invariance

3.1 Introduction

One of the primary challenges of robust visual object recognition is the trade-off between invariance to possible distortions, vs. maintaining a representation that is specific to a particular object. Robustness in the form of invariance can exist for a variety of transformations, such as to translation across visual space, or rotation. On the other hand, robustness requires that an object's neural representation only becomes active when that particular object is actually present, and not another, perhaps similar, object. The process of using the features that are present to robustly determine a conjunction's presence (where a conjunction might be as complex as an object) is one component of the binding problem that the brain must solve [Hummel and Biederman, 1992, Treisman, 1999, Elliffe et al., 2002]. It remains unknown precisely how primate vision carries out binding to enable both invariance and specificity, while artificial systems with comparable robustness to primate object recognition do not yet exist. Explaining such robustness with biologically plausible computations therefore remains an open problem.

In the previous chapter, we considered conjunction detection that is sensitive to the spatial arrangement of features, but where these features are processed in a serial manner across several saccade-like movements. In the following work, we will consider spatially sensitive conjunction detection in a parallel setting, as would occur within a given fixation.

We will begin by outlining a possible approach that builds on prior work [Davis, 2010b, Sabour et al., 2017], where a key aim is enabling online translation invariance, that is, generalization to locations in the input space that have not been seen before. In particular, this work will explore the potential significance of category-slot representations for enabling robustness. After a discussion of theoretical principles, we will consider the possible biological basis for the necessary feature representations. This will be followed with a mathematical analysis to predict the parameter space within which spiking neural networks might implement this computation, taking into account both biological and computational constraints.

In Chapter 4, we will directly simulate spiking neural networks to assess the plausibility of the proposed approach, and expand upon the results of the current chapter's analysis. Our results will demonstrate that the proposed computation is feasible in biologically plausible networks, and that it may explain a component of conjunction detection in primate vision. In addition, however, our work will demonstrate that without additional learning mechanisms beyond Spike-Timing Dependent Plasticity (STPD), the computation is likely to represent a more specialized form of conjunction detection that operates alongside standard pooling-based methods.

3.1.1 Typical Approaches to Translation Invariance

We now turn to the question of achieving translation invariant representations, of which there are two common approaches. In the first of these, local conjunction detection operates at every possible location, after which determining that at least one of the local feature detectors is active (such as through a pooling operation), is sufficient to determine that the conjunction is present. In the second approach, local features are themselves invariant (i.e. respond across multiple locations), while conjunction detection is then performed on these invariant features. We note that these approaches are not mutually exclusive, and can be used in combination.

The first of these is a straightforward solution if a conjunction or object has been learned at every possible location. However, this is ultimately limited by the extensive amount of training data that is required, and so feature-detectors are likely to only exist for low-level features [Elliffe et al., 2002, Stringer et al., 2006, Pogodin et al., 2021]. Furthermore, humans demonstrate the ability to perform online translation invariance for complex objects they have not seen before. That is, given a novel object learned at one location on the retina, humans can recognise that object at a different (often significantly displaced) location of the retina with high accuracy. Importantly, this has been demonstrated in settings where objects differ based on the configural arrangement of low-level features [Bowers et al., 2016, Han et al., 2020, Blything et al., 2021].

Developing architectures capable of online translation invariance that remains robust to the spatial arrangement of features, and using biologically plausible mechanisms, will

form the central focus of these next two chapters. As will become clear below, the primary mechanisms that are used in current models to enable online invariance are of uncertain biological plausibility, or result in a loss of sensitivity to the spatial configuration of features.

We now return to the first approach described above, namely combining responses over feature detectors present at all possible locations in the input space. To enable such a mechanism to function even at locations where a stimulus has never been presented, machine learning models such as CNNs adopt the principle of weight sharing [Fukushima, 1980, Lecun et al., 1998, Serre et al., 2007]. As a result of weight sharing, feature detection at any location will result in an approximately equivalent response. However, the location of the activated representation, in the retinotopic space of a network's layer, will shift according to the location of the feature in the input space. This property of consistent, but shifting, representations is known as equivariance [Zhang, 2019]. We emphasize that by determining whether a feature detector is active at any location, one can determine whether a conjunction such as an object is present, even if it is at a novel, never before seen position. Pooling across all such feature detectors, through a mechanism such as global average pooling, can thereby provide strong translation invariance in many architectures [Blything et al., 2021, Biscione and Bowers, 2021].

This approach has the advantage that, in principle, conjunction detection can account for the particular spatial arrangement of features (i.e. conjunction detection can be form specific), while online translation invariance is still possible. Importantly however, weight-sharing has for several decades not been viewed as biologically plausible. Interestingly, recent work has suggested that a sleep-wake learning algorithm might enable its implementation in the brain, however this approach requires finely structured lateral connectivity, and the ability of this algorithm to support translation invariance (online or otherwise) has not been directly evaluated [Pogodin et al., 2021]. Furthermore, even in non-biologically plausible CNNs, the equivariance of representations often does not hold [Zhang, 2019], which may contribute to such an approach failing in certain architectures [Biscione and Bowers, 2021].

In spite of the above limitations, it is reasonable for biologically plausible networks to

have learned local feature-detectors for simpler features across the entire input space [Elliffe et al., 2002, Stringer et al., 2006], which can be used for the second common approach to invariance, as we now discuss. In particular, by learning to detect simple conjunctions across all possible locations, low-level neurons can respond in an invariant manner to the presence of these features. Conjunction detection can then be performed on these invariant features. Using lower-level features that are translation invariant should provide higher-level representations with a degree of online translation invariance, without requiring weight sharing [Rosenblatt, 1961, Rolls, 1992, Hummel and Biederman, 1992, Riesenhuber and Poggio, 1999a, Elliffe et al., 2002, Stringer et al., 2006]. A motif frequently used in both the machine vision field [Lecun et al., 2015], and in modelling primate vision [Serre et al., 2007, Yamins et al., 2014] is that of max-pooling, where the activation of an output neuron is set equivalent to the highest single activation of multiple input neurons [Riesenhuber and Poggio, 1999a].

A demonstration of this principle is provided in Figure 3.1.a.i, using simple oriented bars that can combine to form a letter T. As the activity of the pooled bar detectors will remain the same regardless of translations within the receptive field, a T detector is imbued with a degree of online translation invariance to novel locations. Note that while the vertical and horizontal bar neurons might be invariant following the learning of these features at every possible location in the input, it is online invariance of more complex conjunctions (such as the T) that we are concerned with. While a possible solution, this approach can render a system less capable of distinguishing the learned stimulus from nonsense arrangements of the features, or from other objects composed of the same low-level features (Figure 3.1.a.ii), as the lack of spatial sensitivity precludes any information about the relative arrangement of features [von der Malsburg, 1981, Hummel and Biederman, 1992]. It is clear from this that, on the one hand, online translation invariance will be quite limited unless the receptive fields of the low-level features are relatively large. At the same time, the larger these receptive fields are, the more insensitive the conjunction detection process is to the spatial arrangement of features.

Under the setting of low-level features that are invariant to large translations, specificity can partly be rescued by the existence of conjunctions which are unique to a particu-

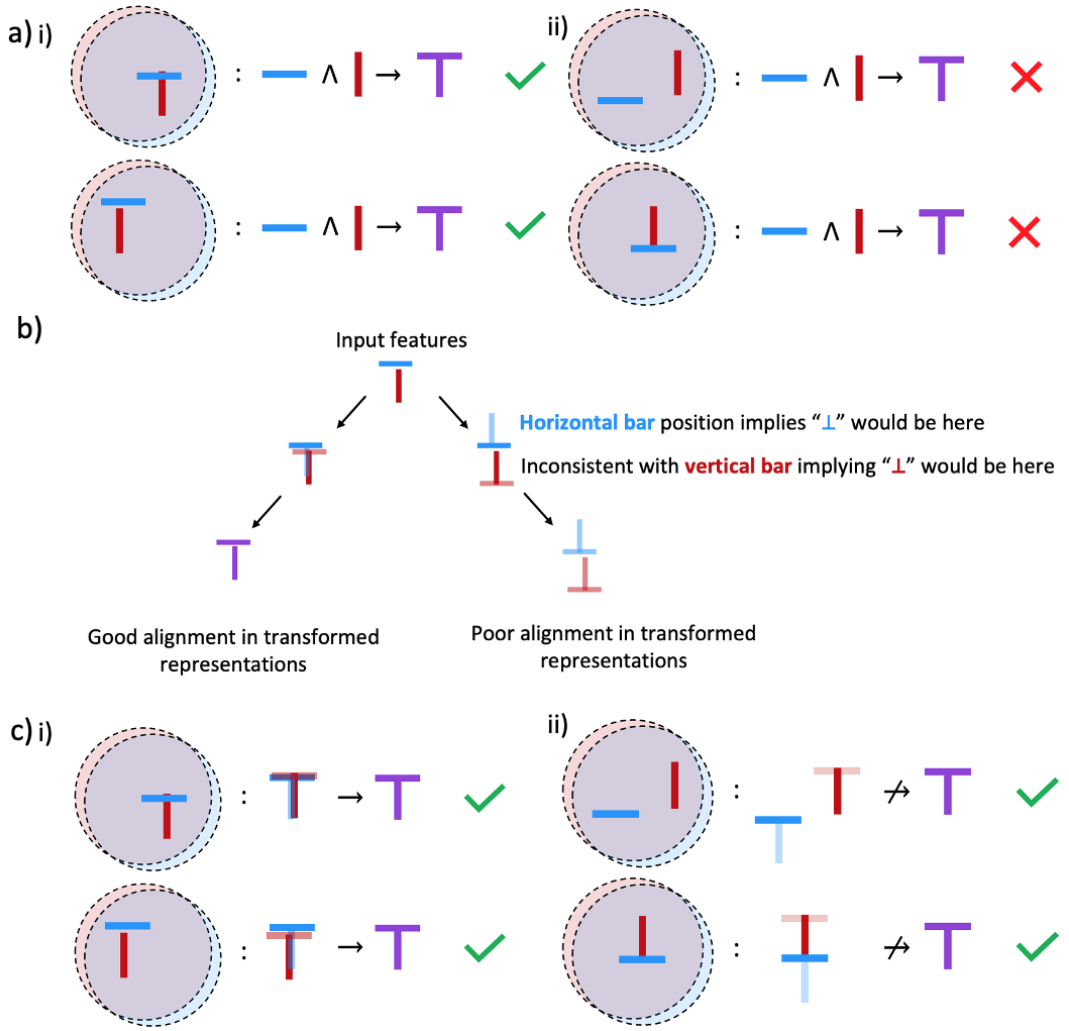


Figure 3.1: *The trade-off of invariance and specificity in object recognition.* a) An algorithm uses the detection of vertical and horizontal bars to determine whether an upright "T" is present. The receptive fields are indicated by the semi-transparent circles. The $\wedge$ symbol is used to indicate the logical AND, while the $\rightarrow$ represents IMPLIES. a)i) The low-level feature neurons shown have learned translation invariance, and thus regardless of where the feature is in the receptive field, the feature representation will be the same. Using invariant low-level features for higher-level computations provides the object representation (here an upright letter "T") with a degree of online translation invariance. a)ii) This same invariance also renders the algorithm potentially vulnerable to misinterpreting arrangements of the low-level features which do not actually correspond to the learned object. b) A system performing capsule-network [Sabour et al., 2017] like computations determines the relative spatial consistency between low-level features. This requires maintaining some information about the spatial 'pose' (such as relative position) of the low-level features. c) The use of the relative spatial arrangement of features enables object recognition that is both i) capable of online translation invariance and ii) form specific.

lar stimulus, thereby enabling what is essentially a bag-of-features detector to maintain specificity to the desired object [von der Malsburg, 1981, Hummel and Biederman, 1992, Rolls, 1992]. In the example shown, this would only be possible if the presence of a vertical and horizontal bar was unique to a single object (e.g. a T), rather than being shared by other stimuli. Elliffe et al. [2002] modeled such an approach as a biologically plausible solution, where local feature conjunctions can often uniquely define objects, while still enabling invariance. Similar results were obtained in Stringer et al. [2006], however the degree of translation invariance observed in these models was minimal compared to that observed in humans [Bowers et al., 2016, Blything et al., 2021], likely owing to the receptive fields of feature neurons being relatively constrained.

CNNs make use of such lower-level, invariant features (typically through max-pooling), in addition to weight-sharing. In spite of the risk that pooling can cause a loss of spatial information, it has been practically observed that this combined approach performs very well on general object recognition, including on large data-sets, and with recognition accuracy that approaches human performance [Krizhevsky et al., 2012, Brendel and Bethge, 2019]. CNNs also demonstrate some degree of online translation invariance, although this is typically very limited without either global forms of pooling, or sufficient pre-training with other images [Biscione and Bowers, 2021, Blything et al., 2021].

While pooling-operations therefore present a possible avenue towards online translation invariance, it is unclear whether they are capable of supporting robust object recognition. In particular, CNN architectures show characteristically non-human responses to distorted images where the spatial relations of features have been lost [Brendel and Bethge, 2019], as well as a non-human bias towards the presence of local features and textures, rather than global object shape [Baker et al., 2018, Geirhos et al., 2019, Malhotra et al., 2020, Geirhos et al., 2021]. Furthermore, pooling as a basis for object recognition is difficult to reconcile with the human ability to rapidly distinguish objects composed of common low-level features [Wang et al., 1994, Gupta et al., 2021], and invariance in CNN architectures may be associated with susceptibility to adversarial examples [Kamath et al., 2021]. Biologically plausible models that adopt the use of invariant inputs have conceded that additional forms of local inhibition would be necessary to ensure the system remains ro-

bust to false conjunctions of low-level feature pairs [Elliffe et al., 2002]. Furthermore, it has yet to be demonstrated through modelling work how general such inhibition is in guaranteeing specificity.

For the above reasons, pooling approaches have generally been viewed as only a partial solution [Serre et al., 2007], and one that will remain vulnerable to an appropriately constructed (non-meaningful) arrangement of features [Hummel and Biederman, 1992]. Together with concerns about the biological plausibility of weight sharing, it is clear that achieving conjunction detection that is both spatially sensitive and robust to online translations represents a significant difficulty.

3.1.2 An Alternative Approach

This challenging trade-off has inspired alternative approaches, both in the computational neuroscience [Hummel and Biederman, 1992, Davis, 1999, 2010a,b], and the machine learning literature [Sabour et al., 2017]. At a high level, these algorithms enable a feature representation to be re-used across a large receptive field, but where additional spatial information is preserved. This combination of factors has the potential to enable downstream recognition that has significant online translation invariance (i.e. due to the large receptive field), while also maintaining sensitivity to the spatial arrangement of features.

We begin with a discussion of such approaches in the domain of machine vision systems for object recognition, where this manifested in the development of capsule networks by Sabour et al. [2017]. In capsule networks, the presence of a feature is encoded alongside information about the spatial particularities of that feature. These spatial details are referred to as the 'pose', and might include information such as the feature's x-y position in the receptive field, its orientation, and its scale. As such, the feature representations are a form of category slots. These features subsequently undergo a spatial transformation (such as via matrix multiplication) to determine whether an object is present, where this spatial transformation uses the pose information to determine the consistency between the arranged features and a learned object representation. An intuitive example of this process is provided in Figure 3.1.b, where the algorithm must determine whether the object is a letter "T", or the falsum symbol ($\perp$). Note that in this case we are not concerned with

rotation/reflection invariance, and thus treat T and ⊥ as different objects, as is the case in the Roman alphabet for the letters b/d/p/q.

We emphasize that this form of recognition determines whether the low-level features are consistent with the presence of a particular object *following a spatial transformation that examines their relative arrangement*. The advantage of this approach is that the system can, in principle, address the opening trade-off. That is, by a feature detector encoding information about pose, and through the subsequent use of this pose information during the recognition process, classification can be invariant to transformations such as translation to a novel location (Figure 3.1.c.i), while avoiding the false recognition that bag-of-invariant-features approaches can suffer from (Figure 3.1.c.ii).

As a concrete example of such processing, imagine two low-level feature detectors for vertical and horizontal bars, which encode i) whether a feature is present in the receptive field (i.e. an invariant response), and ii) information about its position within the one-dimensional receptive field. For the sake of analogy, we describe the features as horizontal/vertical bars (where x^h and x^v describe their respective locations), although in reality they are points in one-dimensional space. Furthermore, we will leave unaddressed for the moment precisely *how* the brain captures information about the positions of the features (relative or absolute). Given that the relative positions of the two low-level features define the object stimulus, then it is straightforward to use this measure as the predictor of the current object (Figure 3.2.a). As the relative spatial arrangement is used, this process is both invariant to translations (Figure 3.2.b), and can robustly encode other objects composed of the same low-level features, such as ⊥ (Figure 3.2.c).

More precisely, we will first define the limits of our two feature detectors' receptive fields as $x^h \in [0, 1]$ and $x^v \in [0, 1]$. Both here and in our later simulations, we assume that all of the considered feature neurons respond invariantly to stimuli in their receptive field, and that the receptive fields exactly overlap with one another. As well as simplifying our discussion, this is motivated by the observation that cortical cells are typically organized in 'minicolumns', wherein neurons encode the same feature, and share overlapping receptive fields [Hubel and Wiesel, 1962, Mountcastle, 1997]. In addition, complex (i.e. invariant) cells in adjacent minicolumns often have overlapping receptive fields, despite

encoding different features [Gilbert, 1977]. Thus, rather than a population of neurons that tile visual space, we will consider two populations of neurons, defined by a large, common receptive field, and distinguished by the features that they respond to.

On the basis that the receptive fields overlap, then the feature offsets of our objects of interest, T and $\perp$, will take up some proportion of the receptive fields (say 25%), within which they can translate. Given $\Delta x = x^h - x^v$, then the object-defining spatial offsets are $\Delta x_T = 0.25 = -\Delta x_\perp$, with some possible spatial tolerance of the relative feature arrangements (e.g. $\epsilon_x = \pm 0.05$). Furthermore, if a feature is not present, then the metric Δx is undefined and the object is interpreted as absent. Figure 3.2.d shows a heat map of the values of Δx as x^h and x^v vary. Note how the two objects trace out distinct Δx values, conferring specificity in spite of the shared use of the receptive fields. Furthermore, the values of Δx are consistent across various absolute positions of the objects, conferring invariance. Importantly, if an object was learned at one location, and the relative-distance encodings generalize to novel locations, then the representation enables *online* translation invariance.

We highlight that the undefined object regions at the edges (i.e. where a +0.25 or -0.25 offset does not exist) are simply a result of the overlapping receptive fields, due to which one feature will eventually fall outside of the covered region. In reality, other neurons would also tile the visual space in these regions.

In principle, the approach to object recognition outlined above solves the opening trade-off, without resorting to the presence of unique, object-defining features. While our work will focus on translation invariance, the potential use of feature information beyond position in a one-dimensional receptive field yields the possibility of invariance to a variety of object distortions [Sabour et al., 2017].

In the machine vision field, ongoing practical challenges associated with capsule networks, such as their computational inefficiency, has so far prevented their mainstream adoption [Barham and Isard, 2019], although they have shown promise in a number of more limited settings [Sabour et al., 2017, Mu and Gilmer, 2019].

While capsule networks have represented an effort in the machine learning field to address

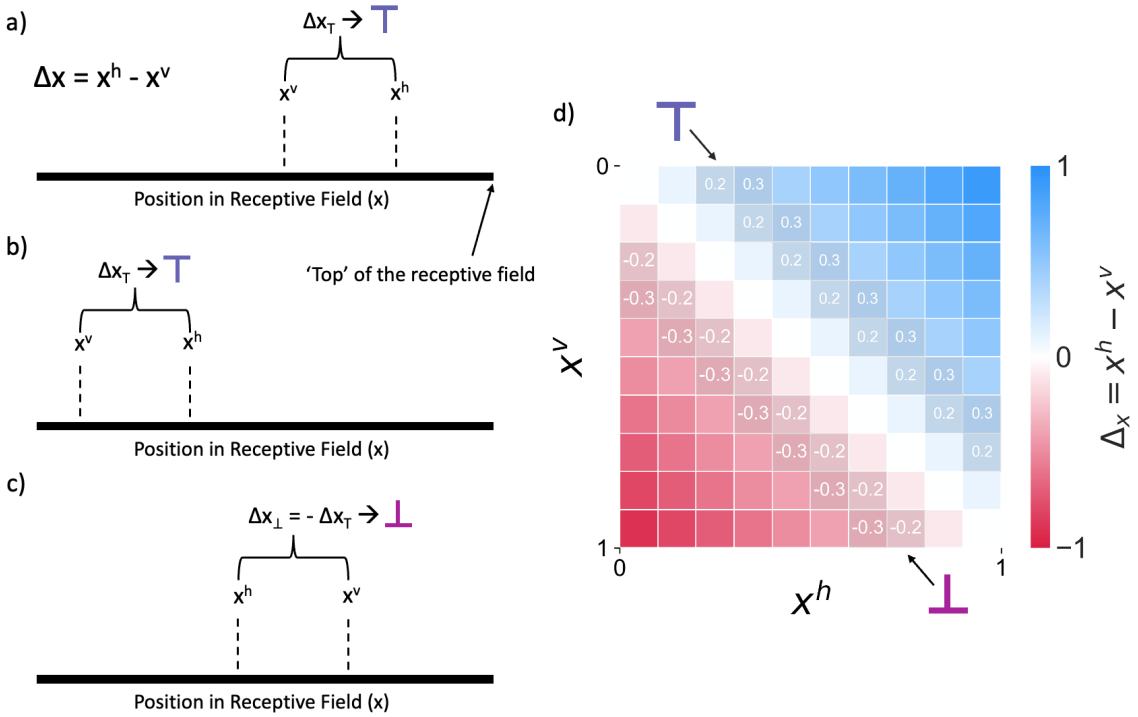


Figure 3.2: *Relative Spatial Information For Robust Object Recognition.* The relative positions of the two low-level features define the stimulus identity in this setup, but if the relative positions are also available to a classifier, they can be used to determine the current object. This encoding of a stimulus at one location (a) is translation invariant, including to novel locations (b), while also supporting the encoding of other objects composed of the same features (c). As the receptive fields overlap, the points are shown in the same one-dimensional space. e) A heat map of the values of Δx as the feature positions vary, with the values associated with the two objects highlighted. Note that the relative spatial position of the features can be visualized by the colour map; in particular, blue indicates that the horizontal bar is above the vertical bar, whereas red indicates the opposite.

the robustness trade-off, related efforts have also taken place in the computational neuroscience literature. Hummel and Biederman [1992] proposed a model where the presence of features and their spatial properties were encoded in distinct neural populations. Temporal synchrony was then used to bind this information, enabling feature identity to be maintained (i.e. an invariant representation), while also preserving other useful information for performing robust recognition, such as the relations among features. While this model addressed the invariance-specificity dilemma, and made analogy to synchrony and neural populations, it was of limited biological plausibility. Furthermore, the model was only able to encode discretized feature properties, such as the relation "above" vs. "below".

More recently, and a critical precursor to our work, Davis [1999, 2010a,b] proposed a model for word recognition, whereby positional information was dynamically bound to nodes representing letter identity (token creation with category slots). The spatial relations of letters were then computed using delay-line computations, where the presence of a word was indicated by the coincident arrival of temporal signals. As this process relied on the relative (rather than absolute) positions of letters, the word recognition method was invariant to novel translations. Promisingly, this model was able to fit a variety of experimental effects observed when measuring human word processing, such as those that occur following the substitution or transposition of letters [Davis, 2010b]. While this work focused on word recognition as a specific computation, Bowers [2017] suggested that such delay-line computations might represent a mechanism in the brain to support robust generalization in a variety of settings. Learning of the appropriate delays lines, it was proposed, could then take place through myelin plasticity (i.e. learning-dependent changes to myelination, and therefore conduction delays). Bowers and Davis [2017] further explored the potential significance of myelin plasticity for such delay-lines, and suggested these computations might be important for tasks including object recognition. Whitney [2001] also proposed a model for word recognition that used feature nodes which carry information about their spatial position. In this case however, the position information was inseparable from the activation level (and therefore the visibility) of the letter, resulting in problematic experimental predictions [Davis, 2010a], and delay-lines were not a component of the word-recognition process.

Despite the importance of the above work, learning in Davis' model was not modelled with biologically established learning rules such as STDP [Davis, 1999], as the mathematical equations governing myelin plasticity have unfortunately yet to be well characterised [Pajevic et al., 2014, de Faria et al., 2021]. Furthermore, the model in Davis [2010b] employed discrete delay-lines during word recognition, and its implementation has not been demonstrated in biologically plausible spiking neural networks. Similarly, Bowers [2017] and Bowers and Davis [2017] did not explicitly model myelin plasticity as a means of enabling the proposed computations. The ability of such a recognition approach to be realised in more biologically plausible networks therefore remains an open question.

In parallel with the above work, members of our laboratory suggested that primate brains might resolve the robustness trade-off in object recognition through the unique temporal properties of spiking neurons. In particular, S. Stringer (personal communication, July 2018) noted that as a feature translates in the input space, the sub-population of neurons it activates will gradually change, recruiting new neurons. This in turn would change the temporal properties of the feature encoding while maintaining an approximately invariant representation via the identity of active neurons in a distributed population. This enhanced feature information (i.e. temporal encoding beyond neuronal identity alone) might provide spatial information for use in downstream classification. Related to these concepts, Isbister and Stringer [2018] suggested that, alongside the invariant identity of a feature encoded in a neuron's activity, 'pose' information as found in capsule networks might be encoded in the spike time of a neuron, relative to stimulus onset. Axonal conduction delays could then serve as a form of spatial transformation of this information, with coincident arrival of signals at an output neuron determining whether the arrangement of features aligns with a particular object or not (J. Isbister, personal communication, July 2018).

3.1.3 Overview of Translation Invariance Chapters

From the above background, it is clear that several lines of work have converged on the potential benefit of category-slot representations for robustness. In spite of this, the necessary conditions for these to serve as a basis for robust object recognition in spiking neural

networks have not been explored analytically or in simulations. Our work therefore builds on ideas from the literature and members of our lab in several ways.

In this chapter, we will begin by describing a mathematical formulation, related to that in Davis [2010b], of how spike times might be used to resolve the robustness trade-off. We will then discuss the possible biological basis of the input features needed to support such a form of conjunction detection. This work will support our analyses in Section 3.6, where we estimate the parameter space in which spiking neural networks might solve the motivating problem of invariance alongside form-specificity. In this approach, we will explore the sufficiency of STDP alone to enable networks performing such object recognition to emerge.

In Chapter 4, we will discuss the biological evidence that neurons carry out delay-line based computations, as well as previous work on object recognition in spiking neural networks. Following this, we will carry out the first spiking neural network simulations to determine whether the desired representations emerge in a biologically plausible neural network.

We note that we will focus on the significance of learning long-term representations (i.e. as opposed to representations in short term memory), and will not consider effects of foveal eccentricity. As such, our work is primarily relevant to the experimental results in Bowers et al. [2016] and Blything et al. [2021]. In our Discussion in Chapter 4, we will also discuss the relevance of our model to experimental investigations of online translation invariance in Rhesus monkeys [Cox and DiCarlo, 2008].

3.2 Relative Spike Times for Robust Object Recognition

To provide an example of how the encoding described in the above introduction would operate in a spiking neural network, Figure 3.3.a shows a simple neural network that could distinguish a T from a $\perp$. In particular, this network takes advantage of the properties of polychrony [Izhikevich, 2006]. We note again that in a polychronous network, neurons fire with fixed, asynchronous temporal patterns that uniquely define features or objects. Afferent spikes travel along axons associated with different delays, enabling these tem-

poral patterns to be processed (and distinguished) by coincidence-detecting neurons with sensitivity at the level of milliseconds [Hopfield, 1995]. Depending on the times at which the low-level features spike, they can encode different stimuli composed of the same feature sets (Figure 3.3.b and Figure 3.3.c), while maintaining invariance in absolute time (Figure 3.3.d). This simple network therefore has the potential to resolve the robustness trade-off we wish to address, although this will depend on the nature of the input spatiotemporal patterns, the precise requirements of which we will now explore.

In particular, while Figure 3.3 portrays an architecture that can recognise objects based on spatiotemporal sequences, this does not address the nature of neural spike-timing needed to overcome the specificity-invariance trade-off. To overcome this challenge, the temporal difference of the input feature spikes should reflect relative spatial information. Note that, unless stated otherwise, we will use the term *relative spike times* to refer to the difference in the times with which a pair of neurons spike.

We begin by defining the absolute spike times of our two neurons of interest in some global reference frame (e.g. relative to stimulus onset), as t^h and t^v, and where these correspond to neurons responding to a horizontal bar and a vertical bar. The definition of the receptive fields of these neurons (i.e. exactly overlapping), and the positions within them ($x^h \in [0, 1]$ and $x^v \in [0, 1]$) remain unchanged from our description in the introduction.

We are ultimately interested in the potential significance of their relative spike times, i.e. $\Delta t = t^h - t^v$, and not in their absolute spike times in the global reference frame, as in cortex these are generally unreliable [Kara et al., 2000]. We note that Δt is a two-dimensional function that depends on the absolute spike time of each neuron.

We will now begin by defining the equation that we will use to model the temporal offset Δt, along with the justification for its form. We will then give a greater mathematical understanding for why the difference in the neural spike times can be described by the equation. In Section 3.3, we will discuss the biological means by which such a formulation of Δt might emerge. Finally, we note that the formulation we present will build on the work of Davis [2010b], although where relevant, we will clarify key differences.

To define the relationship between Δt and the positions of features in their receptive

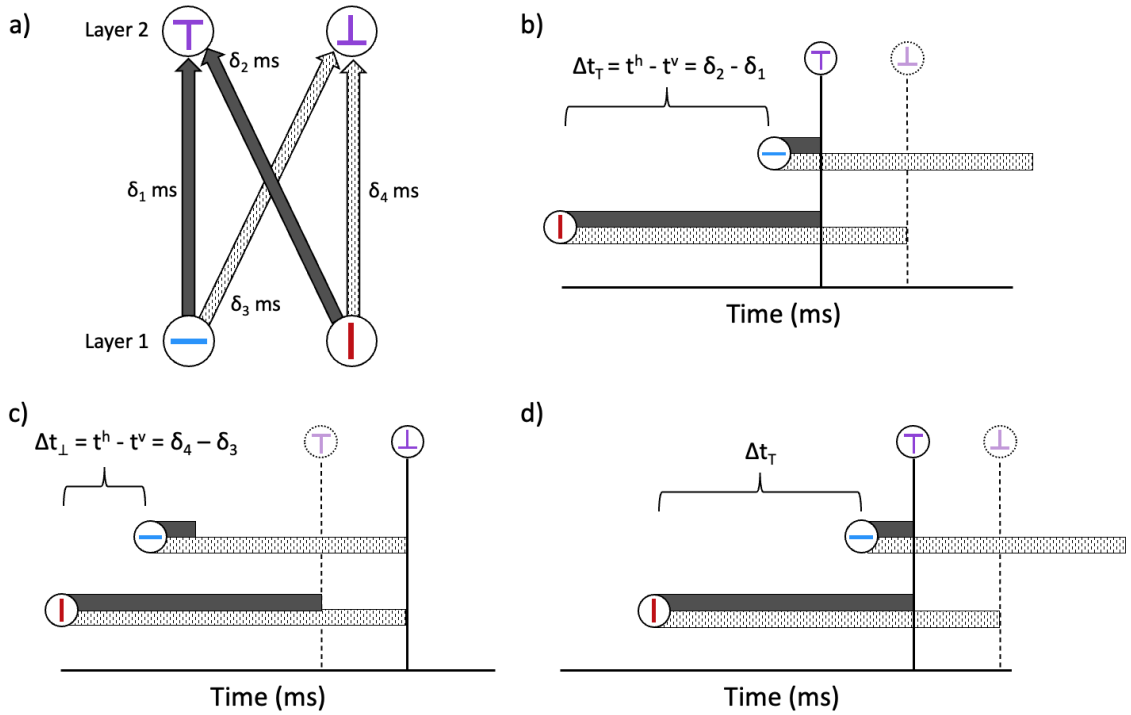


Figure 3.3: *Temporal Information For Robust Object Recognition.* a) A simple spiking neural network that could reliably distinguish a T from a ⊥ while maintaining invariance in the absolute timing of spikes. In particular, conduction delays in the network are such that, if the postsynaptic neurons are sensitive to a narrow window of coincident presynaptic spikes, they will only respond to specific temporal patterns in the lower layer. b) Bold circles indicate the spike times of neurons, while the bars indicate their conduction delays. Faded circles indicate the absence of a neuron firing. The presynaptic neurons spike with a relative delay ($\Delta t_T = t^h - t^v$) such that their signals to the higher representation of a T arrive simultaneously. In contrast, their signals travelling along the textured conduction delays do not arrive at the ⊥ neuron coincidentally, and so it does not fire. c) The presynaptic neurons spike in an alternative spatiotemporal pattern, such that their signals arrive at the ⊥ neuron simultaneously. d) The encoding scheme is dependent on the difference in the firing times of low-level neurons, and so is insensitive to shifts in the absolute spike timing of neurons (e.g. relative to stimulus onset).

fields (x^h and x^v), it is assumed that the relative spike time may change as a function of the position of the features, but that these influences can be captured by independent terms (i.e. a term for each feature, dependent on that feature's position). Such a mathematical model takes the form:

$$\Delta t = \sigma^h(x^h) - \sigma^v(x^v) + C + \epsilon_t \tag{3.1}$$

where $\sigma^f(x^f)$ for any given feature f is an arbitrary (i.e. potentially non-linear) function dependent on the position of the feature in the receptive field, and for which we define $\sigma^f(0) = 0$ for all f. Furthermore, we define C, a fixed offset between the two spike times, and ϵ_t, which represents noise during any given spiking event. Note how, in equation 3.1, the positions of the horizontal and vertical bars (x^h and x^v) will independently influence Δt. We highlight however that this does *not* imply that e.g. $t^h = \sigma^h(x^h)$; as noted, t^h is likely to be highly variable on any given stimulus presentation

On what basis do we assume that the change in the relative spike time can be modelled by independent functions of the feature positions? Firstly, this is motivated by its relationship to one of the primary models which we are attempting to explore the biological feasibility of. In particular Davis [2010b] presented a spatial-coding scheme that is able to fit a variety of behavioural data from word processing in humans. In their model, the relative times with which a pair of neurons fire is a function of independent linear terms, and each one of these terms is a function of the position of a letter in a sequence (we later compare our formulation to theirs in greater detail).

In addition however, the assumption underlying the formulation of equation 3.1 has a biological motivation. In particular, we note that such independent influence would be possible given that neurons can have selective response profiles [Logothetis et al., 1995]. This is especially marked when the features are orthogonal [Pasupathy and Connor, 1999], as in our setting. Indeed, this mathematical model has a natural equivalence to the high-level description provided in Izhikevich [2006] for converting from a rate to a temporal code. This will become clearer in Section 3.3, where we will explore how equation 3.1 might be implemented in biological circuits in more detail.

To understand the nature and motivation of equation 3.1 more clearly, we will now characterise it through a variety of special cases. We begin by defining the case:

$$\Delta t_{x^h=0;x^v=0} = t^h_{x^h=0} - t^v_{x^v=0} = C + \epsilon_t \tag{3.2}$$

where $\Delta t_{x^h=0;x^v=0}$ is an approximately fixed difference in the spike times of the two neurons when both features are present at the origin of their receptive fields (i.e. $x^h = 0$ and $x^v = 0$), determined by the constant C, as well as the degree of noise, ϵ_t. We emphasize again that the absolute spike times of $t^h_{x^h=0}$ and $t^v_{x^v=0}$ are likely to differ substantially for any given image presentation, with respect to the stimulus onset. However, we are interested in the consistency (if any) of the difference in the spike time between the two neurons. It is possible that noise from jitter (ϵ_t) could destroy any reliability of Δt, however we will consider ϵ_t to be minimal until our later simulations. It will therefore be excluded from much of the following discussion.

We now examine a one-dimensional view of Δt when we vary the position (x^h) of a single feature only, while the position of the other feature remains fixed at the receptive field origin ($x^v = 0$). Specifically, we define:

$$\Delta t^h_{x^v=0} = t^h - t^v_{x^v=0} \tag{3.3}$$

where $\Delta t^h_{x^v=0}$ is the spike time difference as the h feature alone is moved, and where this is a one-dimensional case of the two-dimensional function Δt.

We are interested in the reliability and evolution of $\Delta t^h_{x^v=0}$, if it changes at all, as the horizontal bar is translated. For example, translation of the h feature might affect the input firing rates arriving at the h feature neuron. Any influence this will have on neural activity should primarily affect the horizontal bar neuron alone, owing to the specificity of response profiles observed in the visual cortex. Thus, we assume that $t^v_{x^v=0}$ does not vary as a function of x^h, although once again, its absolute value is likely to be inconsistent on any given image presentation with respect to the global reference frame. We therefore assume that this difference in spike times can be captured with a single term, dependent

solely on the position of the horizontal bar feature as follows:

$$\Delta t^h_{x^v=0} = \sigma^h(x^h) + C \tag{3.4}$$

where we note again that $\sigma^h(x^h)$ is an arbitrary function dependent on the position of the h feature in the receptive field, and for which we define $\sigma^h(0) = 0$. As expected, when the horizontal bar is also at its spatial origin (i.e. $x^h = 0$), then:

$$\Delta t^h_{x^v=0} = \sigma^h(0) + C = C \tag{3.5}$$

We can take the same approach for the vertical bar neuron, i.e.

$$\Delta t^v_{x^h=0} = t^h_{x^h=0} - t^v \tag{3.6}$$

Once again, independent receptive fields supports us modelling that the position of the vertical feature only influences the activity of the vertical bar neuron, and so we define:

$$\Delta t^v_{x^h=0} = C - \sigma^v(x^v) \tag{3.7}$$

As before, $\sigma^v(0) = 0$, and thus when $x^v = 0$,

$$\Delta t^v_{x^h=0} = C - \sigma^v(0) = C \tag{3.8}$$

The choice of notation to use a negative sign for the term $\sigma^v(x^v)$, while arbitrary, will become clearer later. In essence however, it captures the notion that similar $\sigma^f(x^f)$ (i.e. similar influences on neural responses for different features) will offset one another, providing the potential for a consistent Δt given a consistent spatial offset (Δx).

Finally, we reorder equation 3.3 as:

$$t^h = \Delta t^h_{x^v=0} + t^v_{x^v=0} \tag{3.9}$$

and equation 3.6 as:

$$t^v = t^h_{x^h=0} - \Delta t^v_{x^h=0} \tag{3.10}$$

We can then combine the definitions provided above to demonstrate the consistency of the system of equations, and return to the more general two-dimensional case when both features can move in their respective receptive fields. Specifically:

$$\begin{aligned}
\Delta t &= t^h - t^v \\
&= (\Delta t^h_{x^v=0} + t^v_{x^v=0}) - (t^h_{x^h=0} - \Delta t^v_{x^h=0}) \\
&= (\sigma^h(x^h) + C + t^v_{x^v=0}) - (t^h_{x^h=0} - (C - \sigma^v(x^v))) \\
&= \sigma^h(x^h) + C + t^v_{x^v=0} - t^h_{x^h=0} + C - \sigma^v(x^v) \\
&= \sigma^h(x^h) - \sigma^v(x^v) + C
\end{aligned} \tag{3.11}$$

noting that $t^v_{x^v=0} - t^h_{x^h=0} = -C$, leaving only a single term for C. As expected, when $x^h = 0$ and $x^v = 0$, $\Delta t = C$, consistent with our beginning definition of C in equation 3.2.

Finally, we can once again consider any noise (which may be completely dominant) in the relative difference of the spike times, providing our final, general equation:

$$\Delta t = \sigma^h(x^h) - \sigma^v(x^v) + C + \epsilon_t \tag{3.12}$$

although in the following discussion, we will continue to consider ϵ_t as negligible.

Equation 3.1 provides us with a general two-dimensional function for how the difference in the neurons' spike times evolves with respect to changes in the positions of *both* features. We emphasize that the formulation is dependent on our assumption that the influence of each feature's position on the relative spike time can be modelled independently. In addition to the justifications we earlier provided for such an assumption, this is motivated by the computational role that such independence will play in potentially

solving the robustness trade-off with which this chapter opened.

We will denote this 2D function encoding relative spatial information in the difference in spike times as a 'spatial-configuration function'. We note again that $\sigma^f(x^f)$ can be any arbitrary function. In Section 3.3 we discuss, with reference to the experimental literature, the influence that these terms might be capturing.

We now turn to the potential significance of the spatial-configuration function for our opening problem of object recognition that is both translation invariant and form specific. Whether the spatial-configuration function has any utility for this task will depend on both the nature of the $\sigma^f(x^f)$ functions, as well as the magnitude of ϵ_t. To aid intuition, we begin by showing spatial-configuration functions where both $\sigma^f(x^f)$ are (implausibly) linear, and ϵ_t can be ignored, i.e.

$$\Delta t = Ax^h - Bx^v + C \tag{3.13}$$

Figure 3.4.a shows the heat-map of Δt given $A = B = 32$, and $C = 0$ (the choice of 32 will become clear later). While unrealistically noise-free, it is clear that this function fully preserves the relative spatial information in the difference in the spike times, hence the resemblance of this heat map to the one shown earlier in Figure 3.2.d. In particular, this spatial-configuration function results in a fixed Δt for a fixed spatial offset ($\Delta x = x^h - x^v$). Later, we will consider more biologically realistic functions for each $\sigma^f(x^f)$, and the challenges that these would present. For now, we highlight that a network with delay-lines and sensitivity to coincidence detection might use the unique temporal offsets in the spike times associated with each feature ($\Delta t \approx 8$ms for T vs. $\Delta t \approx -8$ms for $\perp$) to recognise the particular object presented. Another example of a linear spatial-configuration function with $A = 32$, $B = 24$, and $C = 3$ is shown in Figure 3.4.b.

We note that equation 3.13 is closely related to the formulation provided in [Davis, 2010b] albeit for a two-letter "word", and in the absence of noise. In their formulation, $C = 0$, $A = B$, and the coefficients A and B are largely equivalent to their period P. We highlight two differences:

- Davis [2010b] modelled uncertainty about the position of any given feature with

parameterized Gaussian noise, which was also dependent on the length of a word. It was suggested that this Gaussian noise might reflect the distribution of phase values. We will detail later (specifically Section 3.5.3) how our consideration of noise differs, but at a high level, we will separately consider noise in encoding space (reflected in a non-linear $\sigma^f(x^f)$), and noise in the time at which neurons spike (reflected in jitter).

- Nodes in their system were also associated with an amplitude, which was able to capture the perceptual evidence that a feature is present, and which might be reflected in the number of neurons that become active within a population encoding a feature. Explicitly modelling this is absent from our approach, however in both Section 3.3, as well as our Discussion in Chapter 4, we will explore how our model might integrate information about feature uncertainty.

Finally, we note that our simulations consider objects defined by two salient features (rather than multi-letter words), although we will discuss the incorporation of additional features separately. With these exceptions, our spiking neural network simulations in Chapter 4 will therefore capture much of the mathematical formulation from Davis [2010b] as a special case. This will afford it the benefit of our results both being applicable to this earlier important work, as well the general case we consider, which is the explicit relaxation of the linear formulation of the functions $\sigma^f(x^f)$.

Returning to our proposed formulation, we note that a spatial-configuration function can be challenging to visualize as it becomes more complex, even if shown as e.g. a three-dimensional surface. To better understand their form, we can individually plot the 1D functions:

$$\sigma^h_{pos}(x^h) = \sigma^h(x^h) + C \tag{3.14}$$

and

$$\sigma^v_{pos}(x^v) = \sigma^v(x^v) \tag{3.15}$$

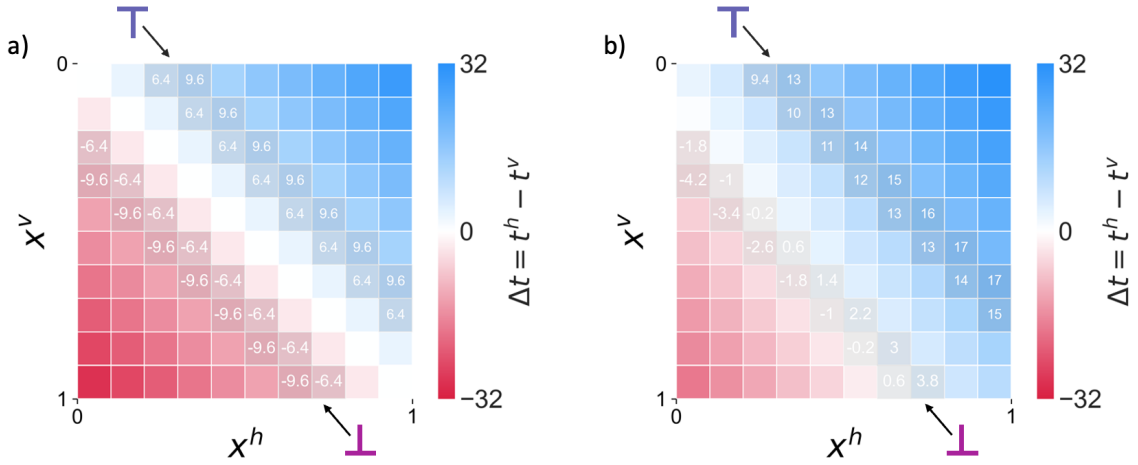


Figure 3.4: *Spatial-Configuration Functions.* Shown are heat maps of $\Delta t = t^h - t^v$ for two different ((a) and (b), see text for parameters) spatial-configuration functions, where they are both defined by idealized linear terms. The regions corresponding to T and $\perp$ are highlighted. Note that for Δt values to be useful, they need not have different signs for different objects. For example, approximate offsets such as $\Delta t = 1$ms vs. $\Delta t = 13$ms for two different objects can still be useful for classification.

which we will term "position functions". Linear examples with a common gradient are shown in Figure 3.5a-b. By only including the constant offset C in the first position function, it is possible for the viewer to visualize an approximate Δt given a particular x^h and x^v simply by examining the graph (Figure 3.5c).

Visualising the position functions has an additional utility, as they represent the underlying process that is modulating the relative spike times of the neurons, providing a sense of the biological plausibility of these processes (e.g. adherence or deviation from strict linearity). We term these position functions as they indicate the transformation of spike times based on the position of a feature. Note however that, except in special cases, the absolute positions of the features cannot be directly read out from Δt. That is, Δt does not uniquely specify x^h and x^v.

In addition to visualising the position functions, one can derive one-dimensional slices of the spatial-configuration function $\Delta t = \sigma^h(x^h) - \sigma^v(x^v) + C$, where the spatial offsets $\Delta x = x^h - x^v$ are fixed to the amounts corresponding to a particular object (e.g. $\Delta x_\perp = -0.25$). This provides the difference in the spike times of the two feature neurons that occur as the absolute position of a particular stimulus changes, i.e. what we will term a 1D 'object' function. Substituting the receptive field positions using a constant spatial

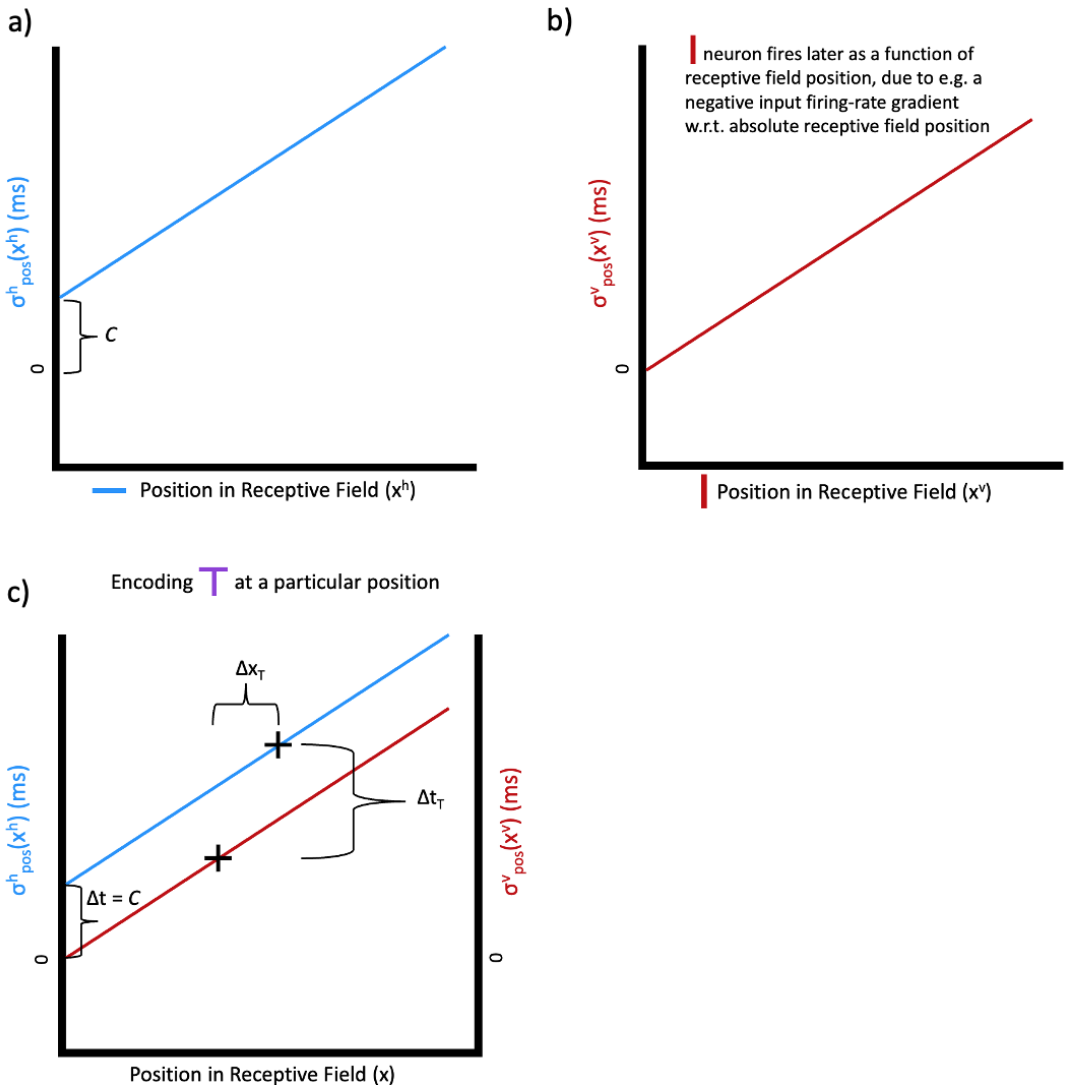


Figure 3.5: *Linear examples of position functions.* The evolution of linear position functions with respect to the position of a horizontal (a) and vertical (b) bar feature. In (b), we provide the example that this process reflects the vertical bar neuron firing later. To understand this, we note that, from our earlier equations, $t^v = t^h_{x^h=0} - C + \sigma^v(x^v)$; thus as $\sigma^v(x^v)$ increases, so too will t^v. c) Both functions are shown superimposed. Given the positions of two features, the difference in the position functions can provide the subsequent temporal offset (i.e. the relevant value of the spatial-configuration function). In this case, the temporal offset Δt is shown given the T object.

offset, and noting that $\Delta x = x^h - x^v$, then :

$$\Delta t_T = \sigma^h(x^v + \Delta x_T) - \sigma^v(x^v) + C \tag{3.16}$$

$$\Delta t_\perp = \sigma^h(x^v + \Delta x_\perp) - \sigma^v(x^v) + C \tag{3.17}$$

Figure 3.6 includes such object functions, where the particular function line corresponds to a particular object's spatial offset. The idealized linear functions in Figure 3.6 demonstrate how information in the difference in spike times can form the basis of object representations that are both invariant to online translations, as well as specific, despite the existence of other objects that are composed of the same constituent features. In particular, as features move in the receptive field with a fixed offset Δx (corresponding to a particular object), the temporal offset Δt will also remain fixed. If this Δt is learned and read out by a coincidence detecting neuron following delay-line offsets, then the representation will be invariant to the absolute position of the object, including to novel locations. At the same time, if the positions of the features relative to one another change (e.g. Δx now corresponds to a different object), Δt *will* change, potentially enabling a neuron to distinguish different objects by their different spatiotemporal patterns, even though they are composed of a common set of features.

Before continuing, it is worth emphasizing that we explore relative spike times where our only concern is the difference in spike times composing an assembly, as this is the most general case. However, a similar, and largely equivalent form of encoding could exist that uses the peak of population activity (e.g. an oscillating local field potential) as an internal reference frame [Izhikevich, 2006, Havenith et al., 2011]. In either case, we emphasize that the spike times are not relative to stimulus onset.

We now consider in greater detail the form of the position functions, and their significance. In order to encode relative feature positions optimally, the position functions would be linear, with a common gradient. In Davis [2010b] for example, the positions of letters were encoded with a single, underlying linear function, where the quality of information

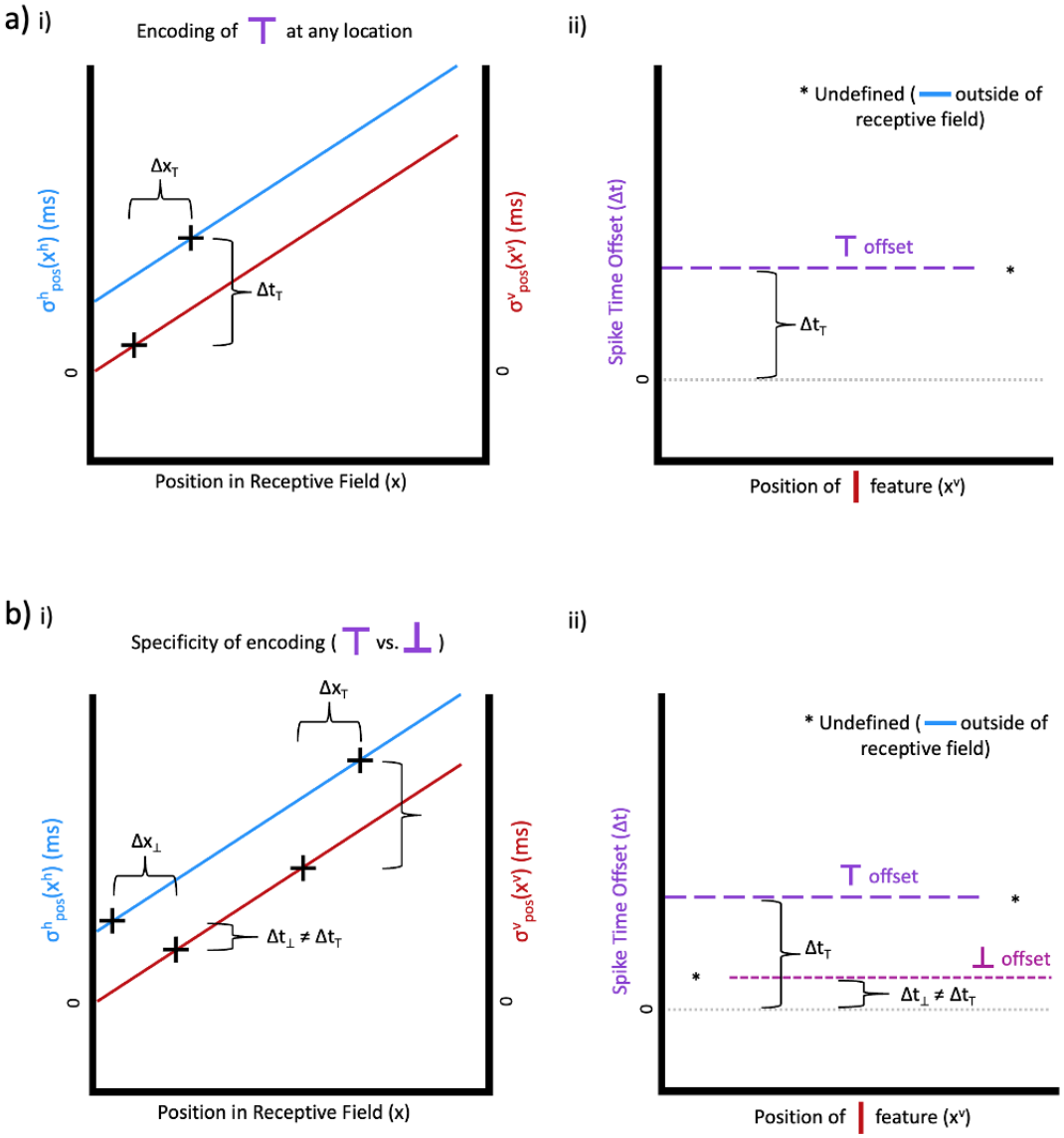


Figure 3.6: *Idealized Spatial-Configuration Function Properties.* To robustly encode an object in the difference in spike timing of neurons, the spatial-configuration function would optimally be formed of linear position functions of the same gradient. a)i) A common gradient ensures that the temporal offset induced by a relative spatial offset of features is consistent, regardless of the absolute position of those features. ii) This consistent offset is reflected in a flat Δt_T as a function of the absolute position of features; a neural network might reinforce synapses associated with conduction delays that offset this particular temporal difference, and these reinforced synapses would support generalization to novel locations. b)i) A non-zero gradient ensures that objects composed of the same features, but for which the spatial arrangements differ, will result in a different temporal offset. ii) Distinct neurons could strengthen appropriate synapses, and thereby uniquely respond to these different (separated) temporal offsets, enabling object recognition that is also form specific.

at any given location was further perturbed by Gaussian noise.

Deviation from this situation could impede generalization to stimulus positions not experienced during training (Figure 3.7a). More generally, spatial-configuration functions, should they exist in biology, are likely to have some degree of non-linearity.

An alternative issue is that the spatial-configuration function may in fact be insensitive to the positions of the features, with consequences for the specificity of the encoding (Figure 3.7b). While it is not possible in this setting to distinguish the two stimuli given their difference in spike times, such processing degenerates to the pooling operation that is typically leveraged in networks. In particular, as long as the low-level feature is somewhere in its receptive field, the feature-neuron will respond, although no additional information about the spatial arrangement of features will be available. As we will discuss later, max-pooling has demonstrated itself as a very capable operation when implemented in hierarchical networks, and so this setting may represent a method for these computations to happen alongside more spatially-sensitive conjunction detection.

While linear position functions with a common gradient are optimal, we highlight that the proposed approach to object recognition does not *require* this, nor will we, in the general case, assume that this holds. Indeed, any monotonic function is likely to provide some utility [Isbister and Stringer, 2018]. For example, sigmoid or exponential functions could also form the basis of encoding information in the difference in spike times, although these may result in sub-optimal generalization to new locations under certain settings (Figure 3.7c).

A specific example demonstrating that two position functions, which are both non-linear and of different form, could still support the robust object recognition we have discussed, is provided in Figure 3.8, where the position functions are both sigmoid. We include the 2D heatmap, position functions, and object functions of the spatial-configuration function.

If neurons do indeed implement spatial-configuration functions in their relative spike times, it is clear that their likely divergent forms will result in a variety of consequences for downstream processing. Robust object encoding depends on complex interactions between the form of the spatial-configuration functions, as well as other properties of a

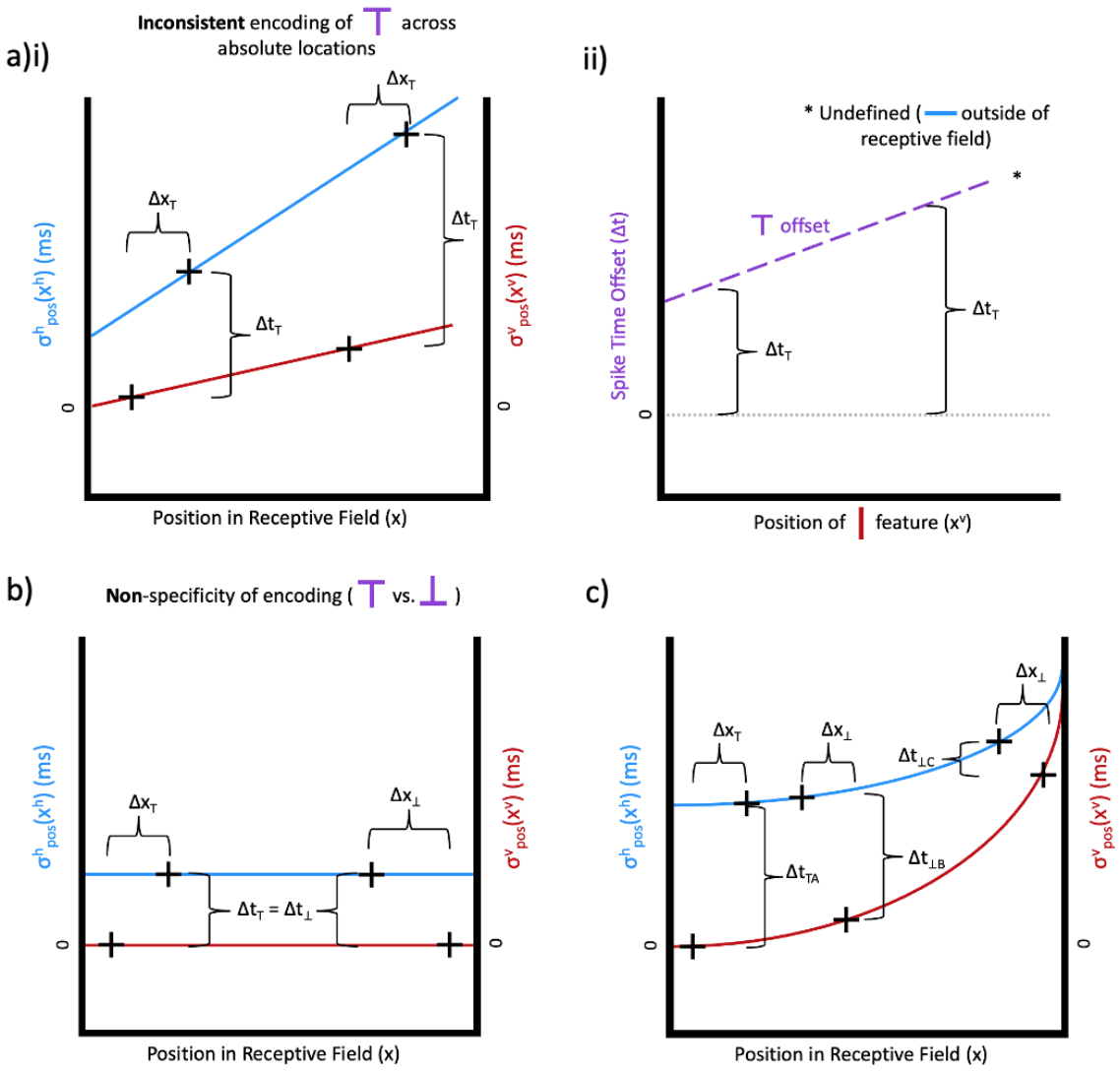


Figure 3.7: *Classes of Position Functions.* a) i) Even if the position functions are linear, inconsistent gradients can cause fixed spatial offsets (e.g. Δx_T) at different absolute locations to result in divergent temporal offsets, potentially compromising generalization to locations not seen during training (reflected in the changing Δt_T in (ii)). b) A possible scenario is that the spatial-configuration function is insensitive to the positions of the features, compromising the ability to distinguish objects composed of the same constituent features. c) General monotonic functions, such as a sigmoid or exponential curve can, in principle, provide the necessary encoding. Note, however, that online translation invariance may not be as robust; for example, there are inconsistent temporal offsets $\Delta t_\perp$ between the absolute locations indicated with subscripts B and C, despite the fixed $\Delta x_\perp$. Similarly, specificity may be reduced in the context of such position functions; in this example, the temporal offsets are approximately the same ($\Delta t_T \approx \Delta t_\perp$) at the absolute locations A and B, in spite of the change in spatial offsets.

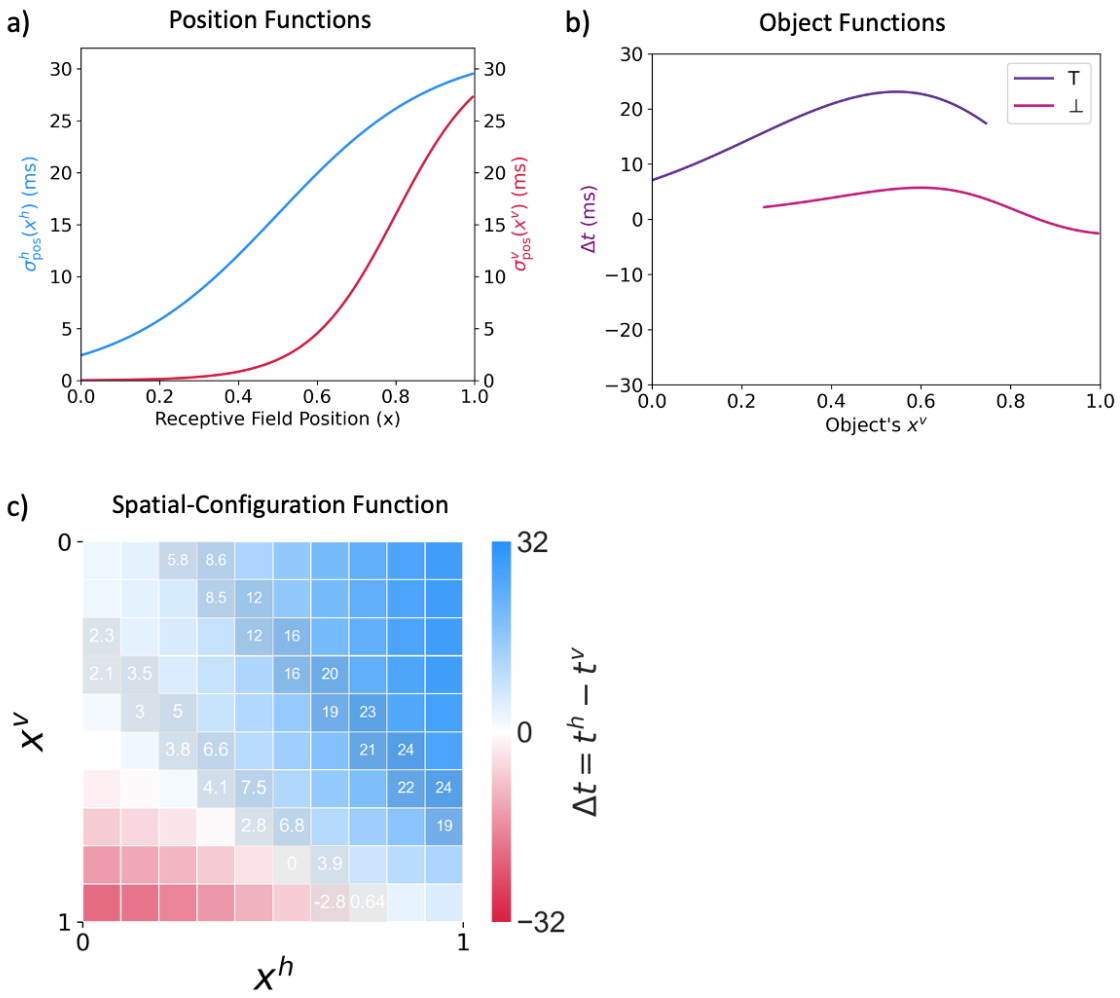


Figure 3.8: *Sample Spatial-Configuration Function with Sigmoid Position Functions.* Shown are the various views of an example spatial-configuration function with sigmoid position functions, including the 1D position functions (a), object functions (b), and the 2D heat-map of the full spatial-configuration function. In spite of the underlying functions not being linear, it can be seen in (b) that the two objects will result in distinct temporal offsets, and that these are broadly consistent (with some tolerance) across various presentation locations. In this particular instance, a neuron that has learned to respond to the ⊥ spatiotemporal pattern might display particularly robust translation invariance, given the flatter (i.e. less variable) object function. At the same time, a ⊥ sensitive neuron may incorrectly respond to a T if it is presented at the far left position ($x^v = 0$). This lack of separation of the object functions is a consequence of the relatively flat $\sigma^v(x^v)$ term in the left-hand region of the receptive field, resulting in a loss of spatial information. Note that in (b), the object functions are plotted relative to the position of the vertical bar feature of the object, although this choice is arbitrary.

network such as a neuron's integration window (defined later), and the consequences of STDP learning. This complexity makes it challenging to directly estimate the non-linear spatial-configuration functions that are sufficient for robust object encoding. As such, our work will use simulations to assess both the conditions under which object recognition based on these functions can operate in biologically plausible neural networks, as well as the settings under which this fails.

It is worth emphasizing that the preservation of spatial information in the difference in spike-timing of neurons is not an alternative to the form of hierarchical binding described in Eguchi et al. [2018], and we contrast this with the view espoused in Isbister and Stringer [2018]. In particular, while coarse information about a feature (such as its relative location in a receptive field) could be encoded in the difference in its spike time, this does not capture the identities of the features that compose an object, nor their hierarchical relations. As a simple analogy, it is useful to know the approximate spatial arrangement of a mouth, nose and eyes when determining whether a face is present. However, the recognition that a face is present tells us nothing about the nature of the features, such as a bulbous or flat nose, that compose the face.

Finally, it is worth addressing objects composed of more than two features, although we will discuss this again later in our work. It is possible that an object (say a letter "I") might be encoded with multiple spatial-configuration functions, where features a and b have some fixed spatial offset (Δx_{Iab}), as well as features b and c (Δx_{Ibc}). By the transitive nature of these offsets, features a and c will also have a fixed offset (i.e. $\Delta x_{Iac} = \Delta x_{Iab} + \Delta x_{Ibc}$), and thus the presence of the object can be indicated by two of the spatial offsets being satisfied. Then, an output neuron might learn to be sensitive to the temporal offsets induced by two spatial-configuration functions. Re-applying the definition of object functions from 3.16 to the two spatial offsets Δx_{Iab} and Δx_{Ibc}, this results in:

$$\Delta t_{Iab} = \sigma^a(x^b + \Delta x_{Iab}) - \sigma^b(x^b) + C^{ab} \tag{3.18}$$

$$\Delta t_{Ibc} = \sigma^b(x^c + \Delta x_{Ibc}) - \sigma^c(x^c) + C^{bc} \tag{3.19}$$

Once again, whether these spatial-configuration functions actually enable a specific or invariant representation of the object will be dependent on their precise form.

While the above approach might be used to maintain form specificity, it may prove less likely for the desired consistency of the position functions to hold as the number of features continues to grow. An alternative, potentially complementary solution is that the described temporal code could operate *alongside* a coarser rate code. In particular, we have argued that it is beneficial to use information about the precise spatial configuration of features to avoid confusing objects with common features. However, it may be sufficient that only a subset of features are computed in this way, while other features (perhaps the majority) undergo a more typical summation dependent on whether they are present or absent.

We have already noted that the setting of approximately flat position functions captures such a computation (albeit still with a temporal code). At the neural level, a rate-coded equivalent might be implemented by feature neurons projecting weaker weights that activate with high firing rates or bursting activity. These neurons could encode features with a coarse form of invariance, insensitive to precise spatial information. In this case, it is not necessary for the signal of a feature neuron to arrive with some precise temporal offset, as it is likely that some of the many emitted spikes will arrive while the receiving neuron is integrating the more temporally precise signals described above. For such rate-coded features that are common to multiple objects, they might serve the purpose of bringing an output neuron closer to threshold. Subsequently, the coincident arrival of temporally precise signals from appropriate spatial-configuration functions could ultimately induce a spike and determine object specificity.

Such a combined encoding approach is consistent with theoretical accounts that small weights support integrative computations, while larger weights support coincidence detection [Rossant et al., 2011]. Furthermore, this is consistent with the experimental observation that synaptic weights in the cortex have an approximately log-normal distribution,

with sparse, strong synapses playing an outsize role alongside a large number of weak synapses [Song et al., 2005, Buzsáki and Mizuseki, 2014]. At a computational level, a combined coarse and spatially-sensitive code could explain why models based solely on the former, such as convolutional neural networks, are able to predict primate data at both neural [Yamins et al., 2014, Bashivan et al., 2019], and perceptual levels [Jacob et al., 2021], while seemingly remaining an incomplete description. Finally, this might help account for the apparently conflicting arguments for rate vs. temporal signals as the primary encoding strategy of the brain. Our own simulations will indirectly capture this notion of combined computation, as the background inputs are temporally imprecise and associated with weak weights. This will allow them to bring neurons closer to the spiking threshold and enable coincident spikes to ultimately determine whether an object-selective neuron spikes.

3.3 The Emergence of Spatial-Configuration Functions

We now discuss the possible biological basis of spatial-configuration functions. The proposal of encoding spatial properties with differences in spike times is consistent with experimental evidence from single-cell activity in the primary visual cortex of anaesthetized cats [Havenith et al., 2011]. In particular, Havenith et al. [2011] observed that in neurons sensitive to oriented gratings, the relative spike times of an assembly of neurons varied in an approximately sinusoidal manner as a function of the stimulus orientation; here, a sinusoidal function is consistent with encoding orientation, rather than translation. Furthermore, a variety of experiments have found that the hippocampus encodes an animal's position within an environment using the timing of spikes [Harris et al., 2002], potentially using oscillatory inhibition to convert a firing rate code into a monotonic temporal code [Mehta et al., 2002]. It is therefore plausible that the spatial properties of visual features could be encoded in a similar manner [Davis, 2010a].

In this work, we will explore whether biologically plausible neural networks are capable of solving the desired invariance-specificity trade-off, *when the given input features encode spatial information in their relative spike times*. Our efforts are motivated by the question of how feasible such a system is for spiking neurons to implement, and how

robust it is to the various forms of noise that will emerge in a biological system. We will therefore not explore precisely how the feature transformation arises, as the details of this will not affect the conclusions of our analysis. Furthermore, there are a variety of mechanisms by which such transformations might occur. We discuss some of these computational mechanisms below, but it is worth emphasizing that even if these prove to be incorrect, other unknown mechanisms might exist to perform the necessary input-feature transformation. Our approach is therefore similar to that taken in Davis [2010b], where the modelling performed was independent of the precise mechanism by which the spatial encoding of letters emerged. The mathematical details of how we model the spatial-configuration functions, including the settings where position functions become increasingly non-optimal, will be provided under our Methods in Section 3.5.3.1.

There are a variety of computational means by which spatial transformations such as translation might be encoded in the relative timing of neurons. Isbister and Stringer [2018] suggested that the diversity of axonal conduction delays present in the brain could convert a spatial dimension such as receptive field position to a temporal code. Specifically, an invariant neuron will typically receive many afferent inputs, each encoding that feature at different locations. As a stimulus translates across the retina, different afferent neurons would become active, but with some potential for overlap in these neurons at any given time. Each one of these neurons would synapse on the invariant neuron with a different conduction delay, and as each position in the input would activate an overlapping subset of these neurons, the spike time of the output neuron relative to the stimulus onset should vary smoothly as a function of translation. While an interesting concept, it is not clear how this form of encoding would promote a linear or monotonic relationship between a feature's position and its spike time. Furthermore, such a mechanism implies that the spikes would occur with a fixed delay relative to the stimulus onset, however evidence of such spike-time precision is largely restricted to pre-cortical visual processing [Berry et al., 1997, Reinagel and Reid, 2000].

In Davis [2010a], it was suggested that a simple architecture could perform a rapid serial scan over an input such as a word. The sequence of letters derived from this serial scan could then be encoded in the spike times of neurons relative to an oscillatory cycle.

Furthermore, the number of feature-specific neurons that become active would encode information about the evidence of a stimulus' presence. This approach has the advantage that there is a clear demarcation between how positional information is encoded (relative spike times) vs. the perceptual evidence of a feature. As already noted, a model based on such an encoding strategy is able to predict a variety of phenomena from experimental results on human word processing [Davis, 2010b]. While this might indeed represent one method employed by the brain, the development of weights in this network requires that the scanning occurs in a consistent direction (corresponding to the direction, such as left-to-right, that readers of that language follow). This requirement could limit such an approach to more diverse objects, where fixation patterns can be largely arbitrary [Foulsham and Kingstone, 2013a].

3.3.1 Monotonic Firing Rates

More generally, a temporal code could emerge due to two underlying mechanisms. The first is that the firing *rate* of low-level neurons encoding a particular feature could vary monotonically as a function of the dimension of interest (such as position in space), and thereby influence the time with which downstream neurons spike. Whether such modulation of firing rate is local to individual neurons, or manifests across a population of neurons, neurons receiving such inputs will tend to fire (i.e. reach threshold) sooner when there is a stronger input [Whitney, 2001, Mehta et al., 2002]. Such early or late firing would be relative to other neurons if their membrane potentials are correlated before the signal is received (Figure 3.9). Indeed, Izhikevich [2006] argued for precisely such a mechanism as a plausible means of converting from a rate to a temporal, polychrony code. We highlight that the accumulation of sub-threshold voltage changes to convert from a firing rate to a temporal code also bears similarities to mechanisms (albeit on different time scales) that can underlie decision making under noisy sensory information [Groschner et al., 2018].

The necessary firing rate functions might have a variety of sources. Within individual neurons, eye position gain fields are well established, modulating the firing rate of a neuron as the position of the eye varies continuously [Zipser and Andersen, 1988, Trotter and

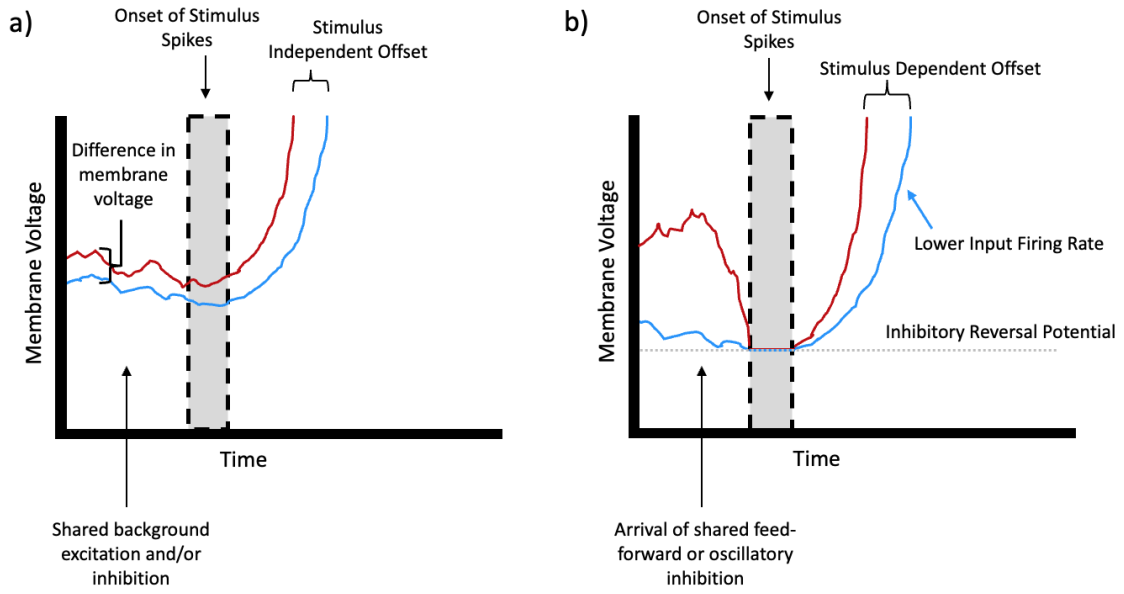


Figure 3.9: *Basis for Temporally Associated Spike Times.* For a firing rate code to be converted to a relative temporal code, an assembly of neurons could have correlated membrane potentials before the arrival of a signal. Under such a setting, the firing rate of the input would determine how quickly a particular neuron reaches threshold, and therefore the timing of the spikes relative to other neurons in the assembly. a) Shared background excitatory and/or inhibitory input would promote membrane potentials that are correlated in time. If the strength of input is overall higher for one neuron than the other, this could also result in a relatively fixed (mean) difference in the membrane potential when no stimulus input is present. In this example, the stimulus firing rate input to the red and blue neurons are approximately the same, but the red neuron fires earlier due to a higher membrane potential at stimulus onset. b) An additional mechanism of transforming firing rates to a temporal code would make use of strong feed-forward inhibition, as described in Izhikevich [2006]. In this case, neurons are driven to a common membrane potential (e.g. the inhibitory reversal potential) by inhibition before the primary excitatory inputs arrive. Here the blue neuron receives a lower input firing rate, and therefore fires later. Such a mechanism is also related to the role for oscillatory inhibition suggested in Mehta et al. [2002].

Celebrini, 1999]. Such gain fields have been observed in neurons in various regions of the ventral stream [Rosenbluth and Allman, 2002] (for an overview, see Lehky et al. [2016]), and the mechanisms by which they can emerge have previously been examined computationally [Navarro et al., 2020]. These gain fields can often be monotonic/planar [Andersen et al., 1985, Zipser and Andersen, 1988], making them well suited for our computations of interest (Figure 3.10.a). Consistent with this, it has previously been suggested that gain fields might play a role in core ventral-stream computations such as object recognition [Graf, 2006]. Similarly, place cells can have skewed firing rate distributions that support conversion to an approximately monotonic temporal code of space [Mehta et al., 2002].

Beyond firing rates that vary *within* particular neurons, there is evidence that firing rate can vary approximately monotonically *across* neurons as a function of anatomical space. This has been observed, for example, as one translates from the upper visual field to the lower visual field in the primate superior colliculus [Hafed and Chen, 2016]. Similar differences in response strength and latency have been observed in visual cortex, either between the upper and lower visual fields, or between perifoveal and peripheral neurons [Hagler, 2014]. This suggests that such gradients are biologically plausible and could represent an anatomical property of retinotopically organized cortex. It would be straightforward, for example, for the diffusion of a biochemical compound to create a population-level gradient in firing rate gain (Figure 3.10.b). Such retinotopic gradients across neurons may have escaped greater experimental reporting if they exist as regional pockets of different signs and magnitudes, yet these could still be useful for downstream processing. Thus, whether in a head-centred, allocentric, or retinotopic space, a variety of sources of the necessary firing rate functions are plausible.

The disadvantage of using monotonic firing rates for a temporal code is the question of how the system can then capture the perceptual evidence that a feature is present, and provide this in a manner that is distinct from spatial information. Ensuring that spatial information and perceptual evidence remain disentangled is an advantage of the scanning network described in Davis [2010a]. We now consider, however, how such disentanglement could be implemented by the mechanisms just described. We note that these mechanisms could also be relevant for the next possible basis for spatial-configuration

functions that we discuss, namely travelling waves.

In one approach, the firing rates of the monotonic-response neurons (i.e. Layer 0 neurons) are largely independent of the perceptual evidence of a feature, so long as the evidence is above a sufficient threshold. The response of the Layer 1 neurons, however, could be gated by an inhibitory circuit, where this influence is dependent on the perceptual evidence of a feature. Such influence could determine the probability that a Layer 1 neuron actually emits its spike. Note that such inhibitory influence need not operate with precision at the level of individual neurons, instead impacting activity across a cortical mini-column. This would result in a similar effect to that in Davis [2010a], where the number of feature neurons that are active would reflect the perceptual evidence of a feature. While we do not explicitly simulate this, we consider in our Discussion in Chapter 4 how such information could be successfully integrated by the network we consider.

Alternatively, we have already noted that coincidence detection and integration can operate alongside one another as a result of the varying effects of strong and weak weights. As such, it may be that the Layer 1 neurons described above are solely responsible for encoding information about the spatial arrangement of features. Firing-rate neurons with weaker, more widely projecting synapses could then encode the perceptual evidence of a low-level feature. If their activity is low, a receiving neuron encoding e.g. a T conjunction would have an insufficient membrane potential to reach threshold, regardless of whether the spikes associated with a spatial-configuration function arrive simultaneously. As such, the firing rate inputs would serve a permissive influence, signifying the sensory evidence of the features that encode an object. This would thereby remove the need for the Layer 1 neurons to simultaneously encode perceptual evidence and spatial information.

3.3.2 Travelling Waves

An alternative mechanism to the above firing rate modulation is travelling waves. In particular, a travelling wave of neural activity (such as a local-field potential) could encourage neurons to fire with different delays relative to one another. If such waves were to travel along dimensions of interest, then the sequential activation of features could be used in downstream processing. Indeed, such spatially consistent travelling waves have been re-

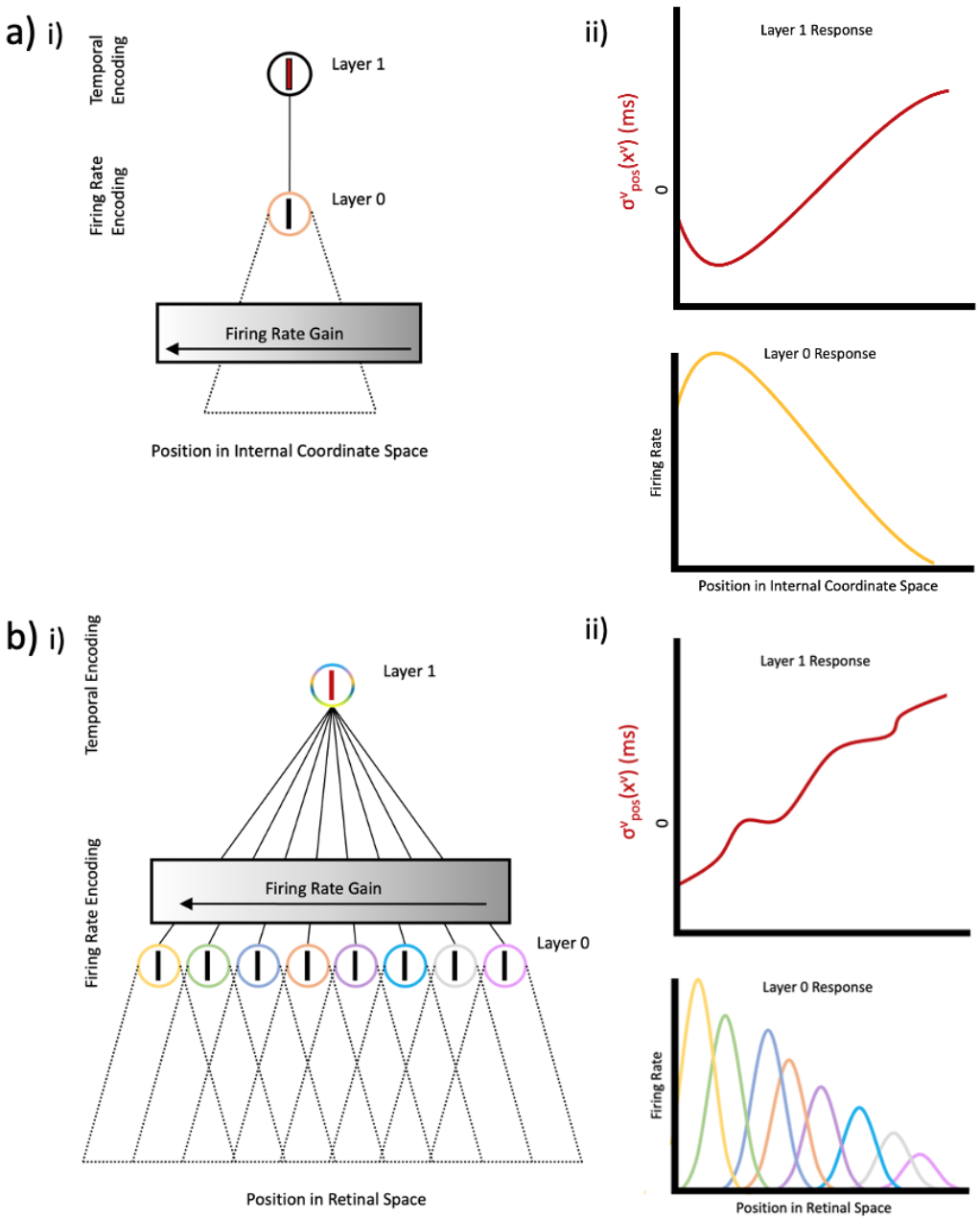


Figure 3.10: *Firing Rates as a Basis for a Temporal Code.* Firing rates that are a function of space could form the basis of a temporal code. a) Neurons can have approximately monotonic firing rate responses to an internal coordinate space, such as is observed with eye position gain fields. Shown is a simple network (i) where such a neuron projects to a downstream neuron, and the influence this might induce through a position function (ii). b) An approximately monotonic evolution of firing rates could also be implemented across a population of neurons, rather than being local to an individual neuron. Here a simple network (i) is shown, where mechanisms such as a biochemical gradient could encourage the modulation of firing rates as a function of retinal position, resulting once again in changes in spiking timing through the position function (ii).

ported under a variety of settings. Most notably, Zanos et al. [2015] observed that in area V4 of macaques, saccades towards stimuli triggered travelling waves originating from the fovea. Importantly, there was an associated correlation between spike-time latency and eccentricity. Given the retinotopic organisation of cortex, as well as the lateralization of visual hemifields [Whitney, 2001, Lavidor and Walsh, 2004], such activity could result in a monotonic function relating eccentricity to relative spike timing. Such influence has been observed to preferentially affect neurons encoding the same feature [Nauhaus et al., 2009], which could assist with the formation of downstream neurons that invariantly respond to a feature, while also encoding spatial information in their relative spike time. Finally, travelling waves have been observed to consistently occur along spatial directions other than foveal radiation [Zhang et al., 2018, MacDowell and Buschman, 2020]. This could enable the encoding of other, orthogonal spatial dimensions through this same mechanism.

It is worth emphasizing that with the above mechanisms, the temporal sequence of neuronal firing is relative to some internal signal, such as an oscillation of local-field potential, or the activity of other neurons forming an assembly [Havenith et al., 2011]. This is thus distinct from neurons spiking at a fixed time relative to a stimulus onset, for which most evidence is restricted to pre-cortical visual processing [Berry et al., 1997, Reinagel and Reid, 2000], and the precision of which appears to diminish as the visual hierarchy is ascended [Kara et al., 2000].

3.4 Preview of Analysis Results and Methods

We will shortly describe the diverse methods that are relevant to both the mathematical analyses performed in this chapter, and the simulation results of Chapter 4. In order to help motivate the understanding of these methods better, we now summarise the main concepts and contributions of this chapter.

Ideally, some subset of the mechanisms explored in the previous section would result in a linear (or at least monotonic) relationship between a spatial dimension and relative spike timing, and this would be similar for the constituent feature neurons, as shown in Figure

3.5. However, the reality is that these functions are vulnerable to deviations from linearity, the manifestation of which can be captured agnostically by noise. In our following investigation, we will broadly categorise noise into two kinds : *structural* sources of noise, which on the time-scale of one or two stimulus presentations can be considered static, and jitter, which will vary between any given stimulus presentation. The former might manifest, for example, if the diffusion of a biochemical compound induced a gradient of firing rates along a dimension of retinotopic space; it is likely that there will be some noisiness to the firing rate distributions. This will affect the input-output slope (i.e. the position functions), and in particular their deviations from a perfectly linear relationship. Depending on the amount of structural noise, this might result in e.g. a more sigmoidal function, but with monotonicity largely preserved, or it might completely destroy any monotonicity. Jitter, on the other hand, will result from the typical sources of noise in neural spike times, such as uncorrelated background activity, or the stochastic release of synaptic vesicles.

In this work, as well as Chapter 4, we will assess whether a biologically-plausible network can learn to solve the invariance-specificity trade-off given features which encode feature identity alongside a varying degree of spatial information. Novel machine vision algorithms, such as capsule networks [Sabour et al., 2017], have suggested that the use of spatial information may be key to robustness, while high-level computational models operating with such recognition can explain a variety of experimental data [Davis, 2010b]. We will determine how robust such a solution is to multiple forms of noise, and under various biologically constrained parameter values, thereby providing evidence as to whether such encoding could actually take place in the brain. The current chapter focuses on the relevant theoretical questions, while Chapter 4 will implement simulations of spiking neural networks.

To study the effects of structural noise and jitter, we will describe in the below Methods (Section 3.5.3.1) the means by which we explicitly model the spatial-configuration functions, and in particular the use of one-dimensional Wiener processes to model the arbitrary functions ($\sigma^f(x^f)$) outlined earlier in Section 3.2. Beyond affording generalisability, our construction of feature inputs will enable us to precisely control the type and quantity of

noise injected into the system.

We emphasize therefore that how the spatial-configuration functions emerge in a self-organised manner, and the precise neural properties that determine their form, are not areas we investigate directly. While these represent valuable questions for future research, our work will focus on identifying the settings under which STDP is sufficient to learn robust representations from such inputs, including the settings under which this fails.

To summarise the contributions of this chapter up until this point:

- We described a general framework for the use of spatial-configuration functions in achieving object recognition that is both translation invariant and form specific. This work builds on that of Davis [2010b], but provides a formulation that is agnostic to the precise functions that describe the difference in feature spike times.
- We summarised, with reference to the experimental literature, the possible means by which spatial-configuration functions could emerge in the visual cortex.

In the following sections, we additionally contribute the following:

- Through a series of analyses, we establish the parameter space under which the proposed network would be capable of operating. This work takes into consideration both biological and computational constraints, while providing a mechanistic understanding of how this space will change dependent on model assumptions.
- This analysis focuses on the contributions of the drift coefficient parameter, the connectivity of a network, and the integration window parameter. Our simulations in Chapter 4 will confirm multiple results from the conclusions we draw here, as well as contribute additional insights into the properties of the networks we study.

3.5 Methods

In the following, we introduce the core methods that will be relevant to both this chapter, as well as the simulations chapter. Note that while the majority of simulations will take place in Chapter 4, some limited simulations will also be performed for our analyses in Section 3.6. Furthermore, the analysis in Section 3.6 is best understood following

an understanding of the network connectivity, and the input stimuli, that we will directly implement in our simulations. As such, general methods, such as the equations describing our model neurons, as well as the hyperparameters of the networks we will simulate, are covered below.

3.5.1 Network Architecture

An overview of the architecture is provided in Figure 3.11, which we will directly simulate in Chapter 4, and which is closely related to the networks we will consider in Section 3.6.

The network consists of two layers, each composed of 20 excitatory neurons. The neurons in the output layer are modelled as leaky integrate-and-fire neurons [Gerstner and Kistler, 2002], while the input layer consists of neurons encoding the identity and location of a particular feature, as elaborated in Section 3.5.3. Neurons in the output layer must learn to consistently respond to the presence of a particular object in order for the network to solve the desired task. This output layer receives activity via feed-forward, excitatory connections modelled as conductance-based synapses. These connections are plastic, and subject to modification via Spike-Timing-Dependent Plasticity (STDP) [Markram et al., 1997, Song et al., 2000]. In order to promote competition, the 20 excitatory neurons in the output layer send connections laterally to a population of 5 inhibitory neurons (also modelled as leaky integrate-and-fire neurons), which send reciprocal, inhibitory connections back. Feed-forward connectivity is only between the input neurons, and the excitatory output neurons, and is all-to-all. Lateral connectivity exists between the excitatory and inhibitory neurons in the output layer, but not within each population, and thus there is no excitatory-excitatory lateral connectivity. Each lateral connection has a probability of 0.5 of forming between any given pair of neurons. In addition to the connectivity within the local network, each excitatory output neuron receives independent background input of 10,000 excitatory synapses. These excitatory synapses are associated with Poisson spike trains, and the weights of these inputs are weaker than the primary synapses within the network.

Feed-forward connections are associated with random delays drawn from a uniform distribution with the range $[0, 15]$ms. This range of delays was estimated from the upper

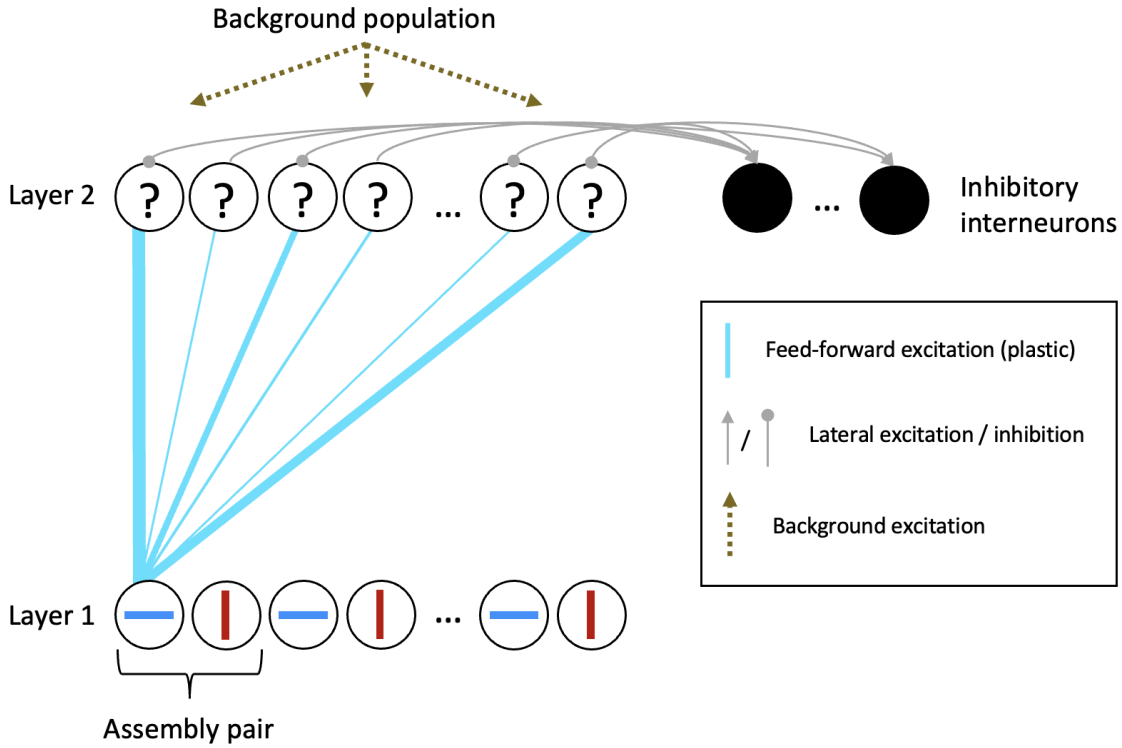


Figure 3.11: *Network Architecture.* The simulated architecture consists of two layers, each with 20 excitatory neurons, and with the addition of 5 inhibitory neurons in the output layer. The input layer consists of 10 assembly-pairs, each encoding a vertical and horizontal bar, and each having temporally associated spikes as described in the text. Only feed-forward excitatory connections are plastic, and the width of connections shown in the diagram signifies differences in conduction delays. For ease of viewing, we only show the feed-forward connections of the first neuron in the input layer (Layer 1), however all input neurons share all-to-all connectivity with the output layer (Layer 2). Lateral connections in the output layer (excitatory-inhibitory and inhibitory-excitatory) have a probability of forming of 0.5. The background input consists of 10,000 Poisson spike trains per excitatory output neuron with a firing rate of 3 Hz each.

limit of local conduction delays observed in Egger et al. [2020]. For simplicity, all lateral connections have a delay of 1ms.

All simulations of spiking neural networks were performed using the Brian 2 simulator [Stimberg et al., 2019] with a numerical time-step of 0.05ms. All simulations results are based on five independent seeds, including the generation of the conduction delays and initial weights of the network. Simulation and analysis code was developed in Python 3.

3.5.2 Differential Equations

The following section describes the differential equations that model the cells, synapses, and learning (in the form of spike-timing-dependent plasticity) of our work. This chapter includes simulations of individual neurons, for which these equations are used. Furthermore, while learning via STDP is restricted to the simulations in Chapter 4, understanding the mathematical basis of STDP, and in particular its temporal sensitivity, will prove useful for the analysis in Section 3.6

3.5.2.1 Cell equations

The evolution of the membrane voltage V of the leaky integrate and fire neurons is described by the differential equation:

$$C_{mem}\frac{dV}{dt} = g_{leak}(V_{rest} - V) + g_{exc}(E_{exc} - V) + g_{inh}(E_{inh} - V) \tag{3.20}$$

The first term on the right describes the leaky behaviour of the neuron towards a resting membrane voltage V_{rest}. g_{leak}, the cell's leakage conductance, and C_{mem}, the membrane capacitance, together determine the rate at which the neuron's membrane voltage decays to the resting state. The second and third terms describe excitatory and inhibitory influence respectively. g_{exc} and g_{inh} describe the excitatory and inhibitory synaptic conductance, while E_{exc} and E_{inh} are the reversal potentials towards which excitatory and inhibitory inputs push the membrane voltage.

As incoming excitatory influence raises the membrane voltage, it eventually reaches a threshold, V_{thresh}, at which a spike occurs. The membrane voltage is then set to V_{reset} and maintained at this point for a period of time determined by the refractory period, τ_{ref}.

3.5.2.2 Synaptic conductance equations

When a neuron receives an incoming spike, its synaptic conductances are modified. The evolution of the synaptic conductances is, in the absence of incoming afferent spikes, described by the differential equations:

$$\tau_{exc}\frac{dg_{exc}}{dt} = -g_{exc} \tag{3.21}$$

$$\tau_{inh}\frac{dg_{inh}}{dt} = -g_{inh} \tag{3.22}$$

These equations describe an exponential decay towards 0, parameterized by the time constants τ_{exc} and τ_{inh}. The update of the conductances following an afferent spike uses the following rules:

$$g_{exc} \rightarrow g_{exc} + \lambda w \tag{3.23}$$

$$g_{inh} \rightarrow g_{inh} + \lambda w \tag{3.24}$$

where w is the current weight of the synapse, and λ is a biological scaling constant, set to be equal in our simulations to g_{leak}.

3.5.2.3 Synaptic learning equations

Learning in feed-forward excitatory synapses is based on spike-timing-dependent plasticity (STDP) [Song et al., 2000], where the change of a weight is defined by the equation:

$$\Delta w = \begin{cases} \alpha_+ e^{(-\frac{\delta t}{\tau_+})} \text{ if } \delta t \geq 0 \\ -\alpha_- e^{(\frac{\delta t}{\tau_-})} \text{ otherwise} \end{cases} \tag{3.25}$$

where δt is the time difference between the postsynaptic spike minus the presynaptic spike. τ_+ and τ_- are constants that describe the exponentially decaying effect of STDP with the decreasing coincidence of pre and postsynaptic spikes. α_+ and α_- are constants whose relative values (along with the respective time constants) determine if the STDP rule has a bias towards long term potentiation or depression of weights.

To efficiently model STDP in our network, we use the same implementation as in Song

et al. [2000]. Specifically, synaptic modification is modelled by two exponentially decaying traces of postsynaptic and presynaptic activity, $M(t)$ and $P_i(t)$ (where i indexes the synapse). In the absence of spikes, these are defined as

$$\tau_- \frac{dM}{dt} = -M \tag{3.26}$$

and

$$\tau_+ \frac{dP_i}{dt} = -P_i \tag{3.27}$$

When synapse i receives an action potential, the presynaptic trace is updated as follows:

$$P_i(t) \rightarrow P_i(t) + \alpha_+ \tag{3.28}$$

Whenever a postsynaptic spike occurs, the postsynaptic trace is updated as follows, resulting in an increasingly negative value for $M(t)$:

$$M(t) \rightarrow M(t) - \alpha_- \tag{3.29}$$

The updates to the weights are then modelled as follows, with a hard border enforced at 0 and w_{max}. Following a postsynaptic spike:

$$w_i \rightarrow w_i + P_i(t) \tag{3.30}$$

for all synapses indexed by i.

Thus, given a postsynaptic spike in the presence of recent presynaptic activity, $P_i(t)$ will be large, and therefore the positive increase in the weight (i.e. long term potentiation) will be large.

Similarly, following a presynaptic spike at a particular synapse i:

$$w_i \to w_i + M(t) \tag{3.31}$$

Note that $M(t)$ will be a large negative value in the presence of recent postsynaptic activity (owing to equation 3.29), and so a presynaptic spike at a particular synapse i following this postsynaptic activity will result in a considerable decrease in the synaptic weight (i.e. long term depression).

3.5.2.4 Hyperparameters

Except where specified in Chapter 4, hyperparameters were generally selected based on previous experimental or computational papers, and are summarised in Tables 3.1 and 3.2. These parameters apply (where relevant) to both the single-cell simulations in this chapter, as well as the network simulations in Chapter 4.

3.5.3 The Modelling of Input Feature Neurons

In the following, we describe how we mathematically model the input features of the network. These results are relevant to an understanding of the general network architecture we consider, as well as the mathematical formulation we use for modelling the spatial-configuration functions. Both of these aspects will be relevant to the results in Section 3.6, and our simulations in 4. Of particular importance is the description of how we will model the spatial-configuration functions using one-dimensional Wiener processes, described below in Section 3.5.3.1.

The input layer of the network we consider consists of assemblies of neurons with temporally associated spike times. Each assembly is composed of two neurons (an assembly-pair), with one neuron encoding feature 'h', and the other encoding feature 'v'. The two high-level stimuli that are to be learned are defined as composed of the 'h' and 'v' features, but in opposite spatial arrangements along a one-dimensional line, as shown in Figure 3.2. While the inputs to our system are abstract spike times (rather than images), we will use the terminology of horizontal and vertical bars, as well as translation in image space, to make the connection between our proposed problem setup and the challenges of

Network Parameters	
Number of Excitatory Neurons, Output Layer	20
Number of Inhibitory Neurons, Output Layer	5
Input	
Number of Excitatory Neurons, Input Layer	20
Number of Stimuli	2
Number of Background Activity Neurons	10,000
Firing Rate of Background Activity Neurons	3 Hz
Saccade Duration	50ms
Presentation Duration (T^{interval})	100ms
Drift Coefficient (μ)	[0, 16, 32]
Diffusion Coefficient (σ^2)	[0, 10, 20, 40, 80, 160]
Jitter (s)	[0, 0.5, 1, 2, 4, 8]ms
Connection Probability	
Excitatory Synapses from Input to Output Excitatory Neurons	1.0
Lateral Excitatory Synapses to Inhibitory Neurons	$0.5^{\bullet}$
Lateral Inhibitory Synapses to Excitatory Neurons	$0.5^{\bullet}$
Neuron Parameters	
C_{mem}	214pF*
g_{leak}	18nS*
τ_{ref}	1.0ms*
V_{rest}	-81.6mV*
V_{thresh}	-52.5mV*
V_{reset}	-57.8mV*
E_{exc}	0mV*
E_{inh}	-70mV*

Table 3.1: **Network and neuron parameters** Italic variable names indicate the primary hyperparameters that are systematically varied. The following symbols indicate the source of other hyperparameter choices : x^{*} value from Troyer et al. [1998]; $x^{\bullet}$ value tuned, details in Chapter 4.

Synapse Parameters	
τ_{exc}	1.75ms**
τ_{inh}	5.25ms*
Maximum Learnable Weight (w_{max})	[1.75, 2.0, 2.25, 2.47]
Initial Feed-forward Weights	$w_i \sim \mathcal{U}(0.5 * w_{max}, w_{max})$
Background Excitatory Weights	$0.005^{\bullet}$
Lateral Excitatory to Inhibitory Cell Weights	w_{max}
Lateral Inhibitory to Excitatory Cell Weights	w_{max}
Excitatory Weight Scaling Factor, λ_{exc}	$g_{leak} * 1.0$
Inhibitory Weight Scaling Factor, λ_{inh}	$g_{leak} * 1.0$
Feed-forward Axonal Delays	$d_i^{exc} \sim \mathcal{U}(0, 15)$ ms‡‡
Lateral and Background Axonal Delays	1ms
Plasticity Parameters	
α_+	0.04
α_-	$2 * \alpha_+{}^{\bullet}$
τ_+	16.8ms‡
τ_-	33.7ms‡

Table 3.2: **Synaptic parameters** Italic variable names indicate the primary hyperparameters that are systematically varied. The following symbols indicate the source of other hyperparameter choices : x^{*} value from Troyer et al. [1998]; x^{**} value from Häusser and Roth [1997]; $x^{\dagger}$ value from Song et al. [2000]; $x^{\ddagger}$ value from Bi and Poo [2001]; $x^{\ddagger\ddagger}$ value estimated from Egger et al. [2020]; $x^{\bullet}$ value tuned, details in Chapter 4.

visual object recognition as intuitive as possible. In the same manner, we refer to the two high-level objects that must be learned as a T and a $\perp$.

With 20 neurons, the input layer is composed of 10 feature assemblies. As noted earlier, each feature neuron is assumed to cover the same receptive field of the input space, e.g. a large area of visual space, through which the stimuli will be shifted during training and evaluation. Due to this shared input space, presentation at any location will be associated with all of the assembly-pairs spiking. As well as the fact that multiple neurons encoding for a particular feature is biologically plausible, the importance of this redundancy for supporting our computation of interest will become clearer in our later analysis (Section 3.6).

During each presentation of a particular object, each neuron, and therefore each assembly, will spike once, although the time at which an assembly spikes *within* this presentation interval is random. In particular, the approximate time at which an assembly-pair will spike is determined by sampling from a uniform distribution of $\mathcal{U}(0, T^{\text{interval}})$. T^{interval}

is the upper limit of the presentation period, and is set at 100ms, a value within primate fixation duration [Salthouse and Ellis, 1980], and in keeping with the range of onset latencies for neurons in macaque visual cortex [Zamarashkina et al., 2020]. We assume that temporally structured firing is clustered in small assemblies (here pairs) rather than across all neurons. Within the brain, this will occur if the mechanisms generating the conversion to a temporal code operate on a predominantly local/sparse level, and indeed this is likely to be more biologically plausible than all neurons firing in an approximately synchronous wave [Song et al., 2005, Takahashi et al., 2010, Havenith et al., 2011]. Finally, we note that the compositions of the assembly-pairs are the same across both stimuli. Thus, neuron identity alone cannot convey any information about the current input stimulus, as regardless of whether a T or $\perp$ is presented, the same neurons will be active.

3.5.3.1 Modelling the Spatial-Configuration Functions

Following the identification of each assembly-pair's approximate spike time in the presentation window, the relative spike times of the composite neurons are determined. This is derived from the spatial-configuration functions introduced in Section 3.3. These determine when the neurons will spike relative to one another, dependent on the position of the features in their respective receptive fields. To solve our opening task, the spatial-configuration functions would be monotonic, and ideally linear with identical gradients. In reality, however, they may be highly non-linear. More importantly, they may also be predominantly flat.

We explore these possibilities by modelling each term in the spatial-configuration function, $\sigma^h(x^h)$ and $\sigma^v(x^v)$, using a Wiener process (also known as Brownian motion) with drift. The Wiener process with drift is defined as a random process in continuous space, where every increment (as defined in equation 3.32) is normally distributed with mean $\mu * \delta x$ and variance $\sigma^2 * \delta x$. Here μ is the drift coefficient, σ^2 is the diffusion coefficient (not to be confused with the terms of the spatial-configuration functions), and δx is the increment taken in the receptive field as one translates a feature. The random process $\mathcal{D}(x)$ will then describe the evolution of the relative spike time as one translates across the receptive field. In summary, every increment of this process, $\mathcal{D}(x, x + \delta x)$ is defined as:

$$\mathcal{D}(x, x+\delta x) = \mathcal{D}(x+\delta x) - \mathcal{D}(x) \sim \mathcal{N}(\mu * \delta x, \sigma^2 * \delta x) \tag{3.32}$$

where the process begins at $\mathcal{D}(0) = 0$.

This modelling approach allows us to systematically vary two key parameters. The first of these is the size of the drift coefficient μ, which contributes to a non-flat gradient, and therefore the potential for encoding the feature's location. When we explore the significance of a non-zero drift (i.e. a bias in the function), the sign is shared for neurons in the same assembly, as the majority of the biological mechanisms we discussed for a monotonic bias are anatomically-based. When we introduce noise (discussed below), it will be clear that we do not assume that the terms of the spatial-configuration functions are linear or share the same gradient. We finally note that if the drift is 0, the function line will be largely flat. In this setting, the relative time at which a neuron spikes will be insensitive to its spatial location in the receptive field, a possibility that we will explore. Using $\mu = 0$, we will also examine the setting of entirely random $\sigma^f(x^f)$ functions, i.e. where there is some spatial coding, but there is no shared linear bias.

The second key parameter is the Wiener process's diffusion coefficient σ^2, which determines the amount of static noise (i.e. deviation from a perfectly linear relationship). As this noise increases, each function (i.e. $\sigma^h(x^h)$ or $\sigma^v(x^v)$) is increasingly unlikely to show monotonic/linear behaviour. This noise is sampled independently for the two terms of the spatial-configuration function, and so increases in the diffusion coefficient will lead to stronger deviations from the optimal coding setting. Finally, when both features are at their receptive field origins, the temporal difference between the neurons' spike times is sampled from a uniform distribution as $C \sim \mathcal{U}(-3, 3)$ms.

Our choice of a diffusion process to model $\sigma^h(x^h)$ and $\sigma^v(x^v)$ is motivated by the desire to capture potentially non-linear functions with scope for a (parameter-dependent) monotonic bias. We would expect the spatial-configuration functions to be relatively smooth, such as the relationship between retinal location and firing rate in the superior colliculus [Hafed and Chen, 2016], and the response properties of neurons sensitive to eye position [Zipser and Andersen, 1988]. As such, we use a diffusion process with relatively large

step sizes ($\delta x = 0.25$), and assume that the spatial-configuration function is sufficiently smooth that it will interpolate between these points. Note however that this does not prevent the function from taking on non-linear forms, and indeed we include a spectrum of sample spatial-configuration functions from our simulations in Figure 3.12.

In summary, this approach enables us to randomly generate spatial-configuration functions, assuming only that there might be some parameter-dependent bias in the function behaviour (i.e. the drift), but where sufficient noise will produce non-linear functions of various forms. We highlight that each term of the spatial-configuration function (i.e. each position function) is modelled with its own Wiener process. These are also separately modelled for each assembly, such that each assembly will have a different spatial-configuration function.

We note that this approach differs slightly from the modelling of noise in Davis [2010b], where Gaussian noise specified the uncertainty about the position of letters along a linear function, and where the amount of noise was dependent on the length of the word. As noise in each position was independent in Davis [2010b], this modelling will result in the function behaviour transitioning from being linear to highly non-linear as the amount of noise increases. In a diffusion process however, samples are taken relative to previous points, and as such, one is more likely to observe instances of smooth, albeit non-linear functions such as sigmoid or exponential curves. Furthermore, noise was captured with a single term in Davis [2010b], making it difficult to assess the contributions of non-linear position functions, vs. noise from jitter. Through our modelling approach, these effects are separated and can therefore be varied in isolation.

Having defined the Wiener process, we now return to precisely define the generation of spike times. After the spike times of an assembly-pair is determined by their occurrence within the presentation window, their respective Wiener processes and positions in the feature receptive fields are used to determine their relative spike time to one another.

Bringing this together, the spike time $t^{v}_{a,p}$ of the v neuron of assembly a within a particular stimulus presentation p is sampled as:

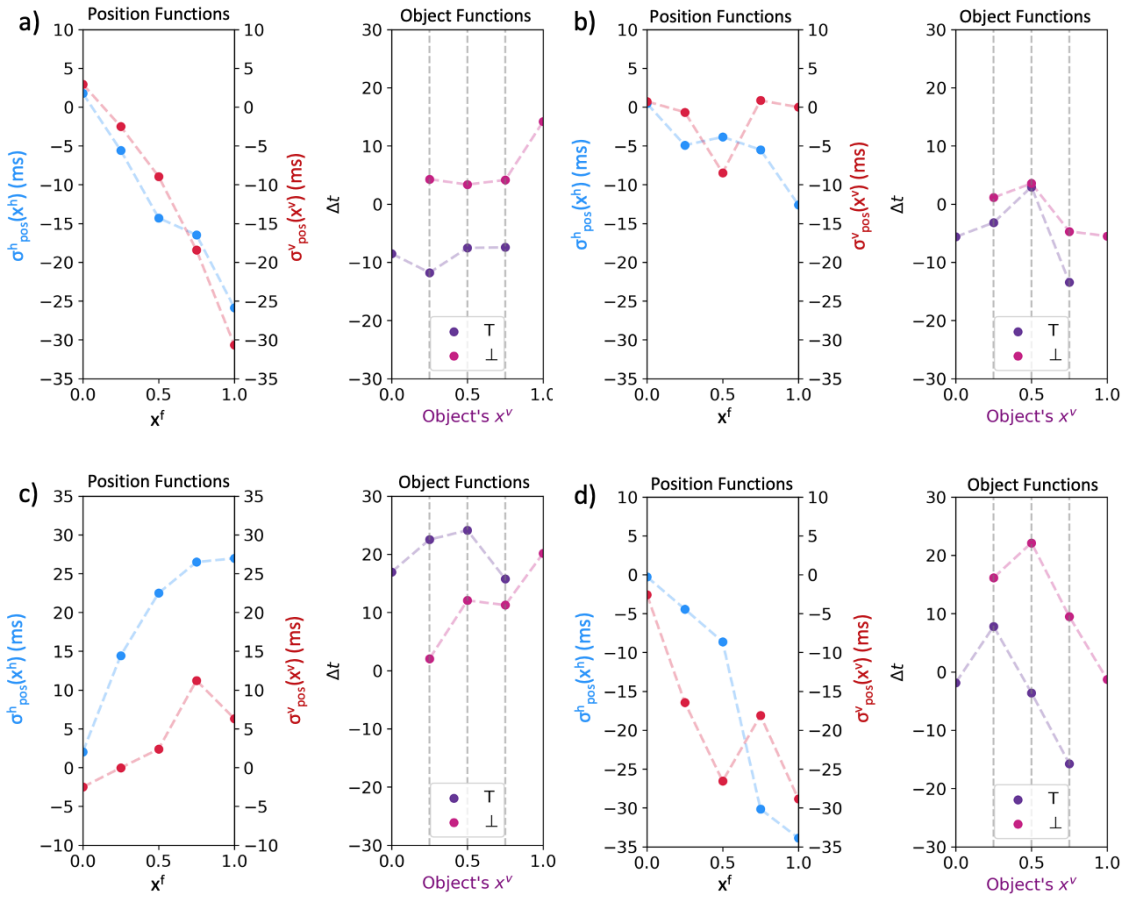


Figure 3.12: *Sample Spatial-Configuration Functions.* Provided are samples of spatial-configuration functions used in our simulations, shown via their 1D position and object functions. As the network's feed-forward axonal delays will need to offset any temporal difference, the flatness of the object functions gives some indication of how well learned representations will generalize across positions. In the setting of an object function with minimal variability, the same delay-lines will result in approximately coincident arrival of signals, regardless of an object's absolute position. Note that the sign of the underlying drift coefficient can be positive or negative. We include a spectrum of spatial-configuration functions (a-d) with varying amounts of drift and diffusion noise.

$$t^v_{a,p} \sim \mathcal{U}(0, T^{\text{interval}}) \tag{3.33}$$

Note that this is random w.r.t. the stimulus onset. The spike time of the $t^h_{a,p}$ neuron is then a result of the difference in its spike time to the $t^v_{a,p}$ neuron. This is determined by the particular assembly's spatial-configuration function Δt, where we model $\sigma^h(x^h)$ and $\sigma^v(x^v)$ with separate Wiener processes, resulting in:

$$t^h_{a,p} = t^v_{a,p} + \mathcal{D}^h_a(x^h) - \mathcal{D}^v_a(x^v) + C_a \tag{3.34}$$

and where, for any given feature f:

$$\mathcal{D}^f_a(0) = 0 \tag{3.35}$$

Underlying this spike-time modelling, recall that each term of the spatial-configuration function reflects modulation of a feature neuron by changes such as a higher or lower input firing rate in certain absolute positions of the receptive field. The actual spike times will therefore depend on both the object presented (T or $\perp$), as well as the position of the object in the receptive field. Both the spatial extents of the receptive fields ($x^h \in [0, 1]$ and $x^v \in [0, 1]$), as well as the spatial offsets corresponding to the two objects ($\Delta x_T = 0.25 = -\Delta x_\perp$), are as described previously.

We further define

$$C_a \sim \mathcal{U}(-3, 3) \tag{3.36}$$

where C_a is the temporal offset between the feature neurons of assembly a, assuming $x^h = x^v = 0$. As a result, even with a drift of zero, or no diffusion noise, there is likely to be a non-zero temporal offset between the spike times of the two neurons.

Finally, jitter is then sampled as follows:

$$t^{f,\text{jitter}}_{a,p} \sim \mathcal{N}(0, s^2) \tag{3.37}$$

where $t_{a,p}^{f,\text{jitter}}$ is added independently to each neuron's spike time defined above (and recalling that f is any feature such as v or h). In our simulations, we will systematically vary the standard deviation (s) of the jitter, as well as the diffusion noise.

Note how $t_{a,p}^{v}$ and C_a are unique to a particular assembly, while the Wiener process terms and jitter are unique to each neuron. Furthermore, while $t_{a,p}^{v}$ and $t_{a,p}^{f,\text{jitter}}$ are sampled for each stimulus presentation, C_a and the Wiener processes represent structural properties of the assembly and are the same across presentations. Finally, we emphasize that the stimulus *identity* does not artificially modify the spike times, i.e. neurons simply spike at the time determined by the locations of the features.

When implementing the above in practice, we correct the range of the uniform distribution in 3.33 using the maximum sampled spike-time offsets for each assembly (i.e. after accounting for jitter etc.), such that the final spike time never falls outside of the presentation window.

3.5.3.2 Presentation of the Input Spikes

We now describe how the sampling of these spikes takes place within the context of training and evaluation. During network training, stimuli in the form of the T and $\perp$ are presented in an alternating fashion. The spike times for every assembly are governed by the above equations, and the particular positions of the vertical and horizontal bar features during a presentation (x^h and x^v). Between each stimulus presentation, a noisy input is also given, where the spike time for each neuron within the presentation window (duration 100ms) is entirely random, i.e. sampled independently from a uniform distribution for each neuron. Note therefore that the firing rate of the input neurons during each presentation is the same regardless of whether the stimulus is T, $\perp$, or simply noise.

During evaluation (i.e. after STDP), no interleaved noise is presented, and the input simply alternates between the two stimuli. Following any given presentation, a 50ms 'saccade' period during which no additional input is provided takes place, primarily in order to simplify the later determination of which network spikes are associated with which stimulus. Afterwards, the next presentation interval begins. In effect, therefore, each input neuron has a firing rate of approximately 6.6 Hz across the simulation, or 10 Hz

during the stimulus presentation. While the timing of spikes within a given *assembly-pair* will be structured for a given presentation, we emphasize again that the timing of an assembly's spikes within the presentation interval (and therefore relative to the assembly's spikes elicited following future stimulus presentations) is random according to the uniform distribution in equation 3.33. Example spike times of the input neurons are shown in Figure 3.13 during a period of training.

We note that the presence of an object is indicated by the spiking of two temporally coincident feature neurons, and the maximum weight of synapses is sufficiently strong so as to reflect this. In other words, if an output neuron at the mean membrane potential receives two coincident input spikes with a weight of w_{max}, it will be driven to spike. While biological neurons typically require the summation of several spikes to trigger an action potential, estimates suggest that as few as three simultaneous spikes can be sufficient [Angulo et al., 1999], and single synapses have been measured to induce postsynaptic potentials as large as 8-10mv [Thomson et al., 1993, Song et al., 2005, Lefort et al., 2009]. As noted in our earlier discussion of higher-dimensional feature sets, coincidence-based features are likely to represent only a subset of the features that drive a neuron to threshold. Other features with weaker weights and high effective firing rates (such as the background inputs that we simulate), can serve the purpose of elevating a neuron's membrane potential such that a small number of strong, coincident signals can drive object selectivity. Our approach is therefore consistent with biologically plausible computations. Finally, the sufficiency of two coincident spikes is consistent with the approach taken in previous computational models exploring coincidence detection, owing to the tractability that this enables [König et al., 1996, Rossant et al., 2011, Kanev et al., 2016].

3.6 Results : Establishing the Parameter Space

We wish to establish whether spiking neural networks can make use of information in spatial-configuration functions to solve the opening task of object recognition that is both robust to online translations, whilst still being able to distinguish objects composed of a common set of features, where the spatial arrangement of these features is what uniquely defines a given stimulus. To do so, we will first establish the parameter space within

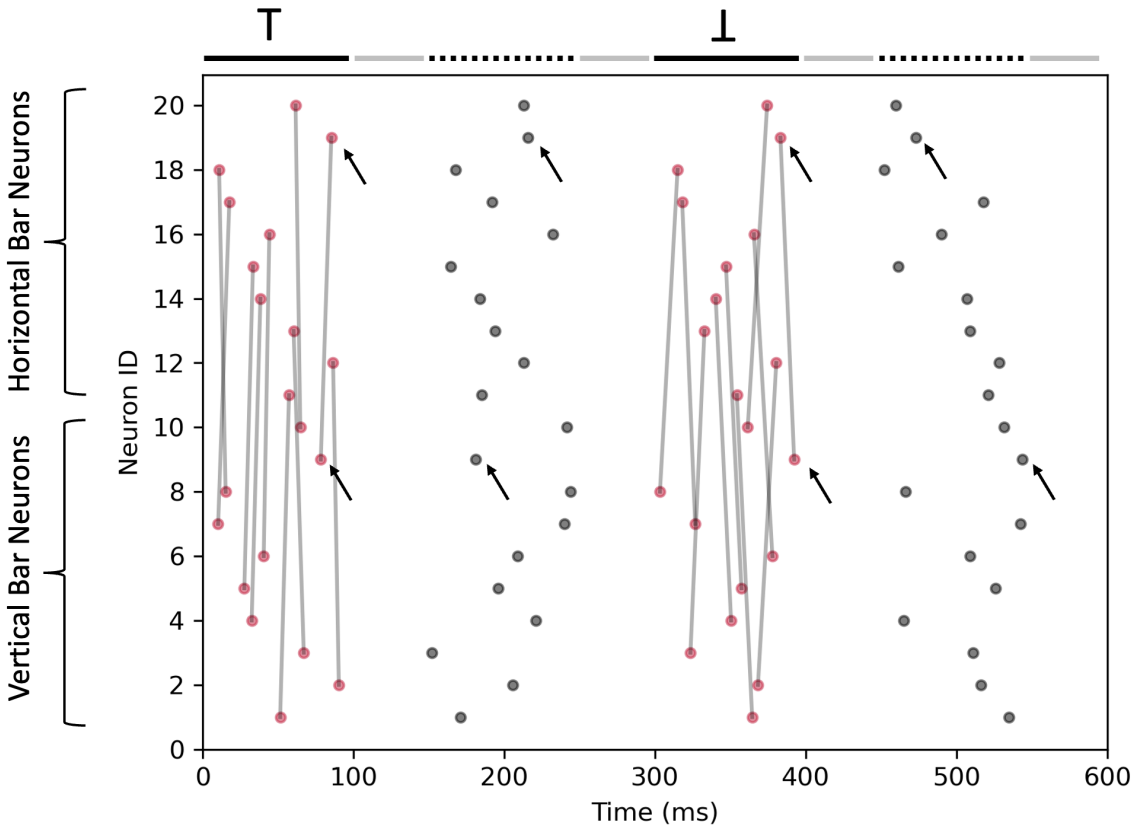


Figure 3.13: *Sample Input Layer Raster.* Provided is a sample raster from our later simulations, showing the activity of the input neurons. Solid bars over the raster indicate a stimulus presentation, while the top grey bars indicate a saccade interval, and dashed bars indicate a noisy (i.e. completely random spike time) input. Transparent lines connecting spikes in the raster plot indicate temporally associated neurons (assembly-pairs), where each assembly is composed of a vertical and a horizontal bar neuron. Note that the compositions of these assembly-pairs are the same across both stimuli, and thus neuron identity alone cannot convey any information about the input stimulus. We highlight a particular assembly pair (arrows), showing how the relative spike times in this case shift subtly for the two different stimuli. Also visible are their entirely random spike times during the noisy stimulus interval. Finally, we note that the timing of any particular assembly pair is random with respect to the other assembly pairs.

which a neural network might actually solve the task, taking into account both biological and computational constraints. In Chapter 4, we will then explore the sensitivity of simulated models as sources of noise are introduced, and the model assumptions that support temporally precise computations are relaxed. As we will demonstrate, multiple factors will interact to create an enclosed region of parameter space within which the proposed computations could feasibly operate.

We emphasize that none of these considerations create hard bounds. Rather, they will enable an informed view of the values that these properties would likely take, should the proposed circuits exist in the brain. This will allow a comparison of the proposed space to current experimental estimates of these parameters. Importantly, this analysis can be revisited as future experimental results emerge. These findings might lead to either the relaxation or further tightening of a particular constraint. In order to enable such future evaluations, we strive in the following results to provide a mechanistic understanding for precisely how the various model properties interact.

3.6.1 The Drift Coefficient and the Temporal Offset of Features

In Section 3.2, we introduced the notion of spatial-configuration functions, while in Section 3.5.3.1, we described how we will model these, both for the following analysis, and in our simulations. We now consider how the form of the spatial-configuration functions will influence the ability of a spiking neural network to develop representations that are both robust to online translations, and sensitive to the spatial configuration of features.

The first important consideration is the magnitude of the drift coefficient μ, and the associated temporal offset of the features composing different objects. As already noted, our spiking neural network (SNN) simulations will explore both non-zero values for the drift coefficient, as well as the consequences of zero drift (i.e. flat position functions). It is important, however, to establish an upper bound on the possible value of μ. In our setting of translation invariance, zero drift results in no information about the position of a feature. While any non-zero gradient is technically sufficient, a larger drift coefficient for a fixed object size also means that the temporal offset of features that compose an object will be larger. This has significance for the robustness of representations, which we will

later explore.

To demonstrate the relationship of the drift coefficient with the temporal offset of features, we will continue to treat the system as if it is noise-free, such that $C = 0$ and $\sigma^2 = 0$, and the gradients of the position functions are the same. These assumptions will enable us to make predictions about the significance of μ, although our simulations will naturally relax these assumptions. In particular, this enables us to define a new term for the magnitude (Δt_{obj}) of the temporal offsets corresponding to each object function, such that $\Delta t_{\text{obj}} = |\Delta t_T| = |\Delta t_\perp|$. In other words, Δt_{obj} is a single term that applies to both objects, and we note again that $\Delta x_T = -\Delta x_\perp$. Turning to the significance of the drift coefficient for Δt_{obj}, we note that with the given assumptions:

$$\begin{aligned} \Delta t_{\text{obj}} &= |\Delta t_T| \\ &= \sigma^h(x^v + \Delta x_T) - \sigma^v(x^v) + C \\ &= \mu(x^v + \Delta x_T) - \mu x^v \\ &= \mu \Delta x_T \\ &= |\Delta t_\perp| \end{aligned} \tag{3.38}$$

Thus, given a fixed object size such as 25% of the receptive field, Δt_{obj} grows linearly as a function of μ. As we will later see, this larger offset is useful for enabling neurons to integrate spikes across a broader temporal window, which can enable greater noise robustness.

Estimating μ from Experimentally Measured Neural Activity While a larger μ, and therefore Δt_{obj}, might be beneficial from a computational point of view, experimental data can be used to estimate a plausible upper bound of μ. In Havenith et al. [2011], they observed that the relative spike times of neurons varied smoothly as a function of the orientation of a drifting grating. The variations in relative spike times that they observed were approximately sinusoidal, consistent with encoding orientation, where one would likely wish to capture the similarity of a grating at 0°and 180°, as well as smoothly return to the same representation following a full rotation (Figure 3.14a.i). Assuming these

relative spike times might reflect position functions, they cover an upper and lower limit relative to other neurons in the assembly of approximately ± 8ms. To estimate the possible range of spike times for a linear (rather than sinusoidal) position function that the brain might implement, we wish to quantify the total change in the spike times that occur as the full receptive field is traversed. As a simple approximation of this arc length, we use the sum of the linear segments in Figure 3.14a.i, i.e. 32ms, where this distance covers the entire receptive field of 360°. In particular, if each section of receptive field is associated with a positive or negative shift in relative spike times of 8ms, these will occasionally change sign in a sigmoid position function, but would all be of the same sign in a linear position function. Given a position within the receptive field (x) that varies from 0 to 1, this monotonic setting corresponds to a drift coefficient of 32ms (Figure 3.14a.ii), which may be of positive or negative sign. In our simulations in Chapter 4, we will therefore assess the performance of the network given a drift coefficient of ± 32ms, as well as ± 16ms (as a more conservative estimate), and 0 (i.e. a flat function in the absence of noise).

Estimating Δt_{obj} from Experimentally Measured Conduction Delays As noted above, a larger drift coefficient, can, for a fixed object size, enable a larger Δt_{obj} (and thereby potentially support robust representations). In spite of this, the data in Havenith et al. [2011] suggests an upper limit of $\mu \approx 32$ms. With an object that fills a quarter of the receptive field, there remains a significant portion of the receptive field where online translation invariance can potentially occur. From equation 3.38, this drift coefficient and object size together correspond to $\Delta t_{\text{obj}} = 8$ms (Figure 3.14b). We now demonstrate how experimentally measured conduction delays in mammalian cortex can provide additional evidence that Δt_{obj} is likely to be bounded from above to approximately 8ms, regardless of the drift coefficient and object size.

In Egger et al. [2020], local cortical conduction delays were estimated in young-adult rodents and found to follow a log-normal distribution with a mean of 3.4 ms (SD $\pm$ 2.3). As we noted in our Methods (Section 3.5), we therefore identified 15ms as an upper limit to the uniform distribution of conduction delays in our model, and we will explore whether, following learning, the distribution of non-pruned synapses (i.e. those with a

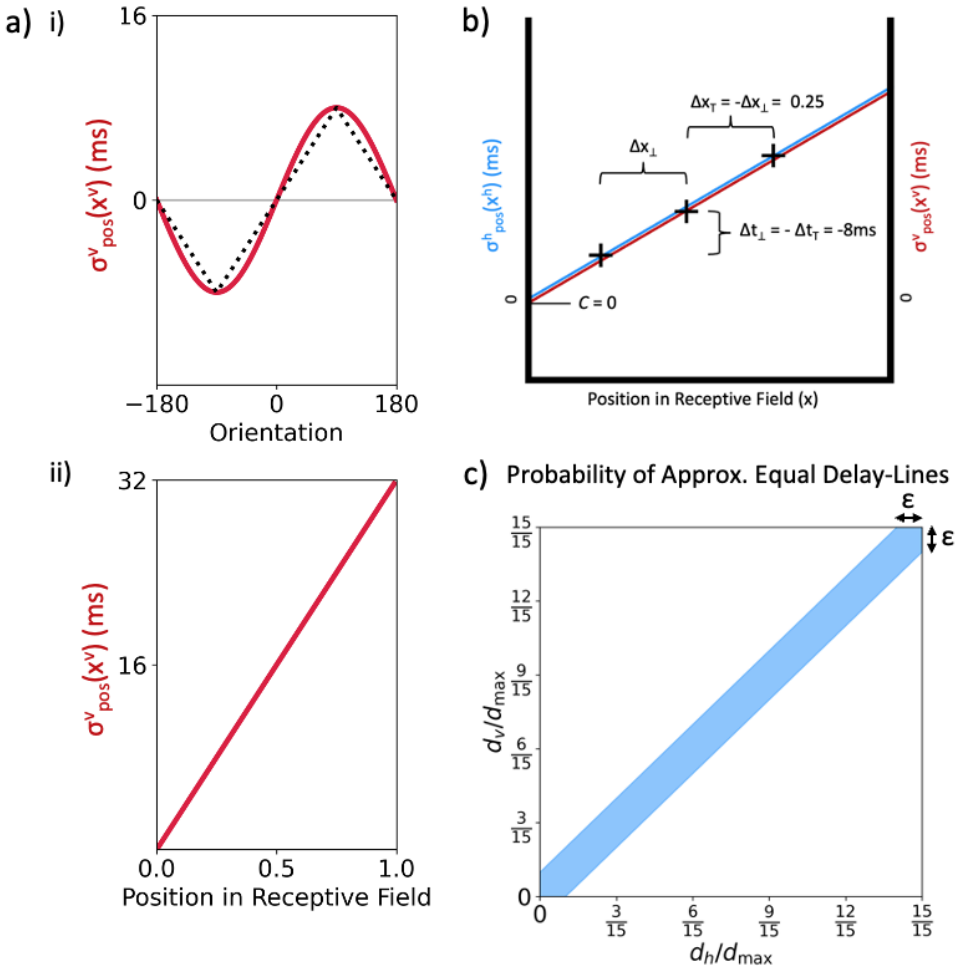


Figure 3.14: *Estimating an Upper Limit on the Drift Coefficient μ and Δt_{obj}.* a) i) In Havenith et al. [2011], the spike times of neurons relative to one another vary in an approximately sinusoidal fashion as a function of feature orientation. These relative offsets fall within a range of around ± 8ms. Assuming these might reflect position functions, we plot an idealized sigmoid function corresponding to this range. The dotted line traces out a simple linear approximation of the arc length, corresponding to 32ms. ii) A linear position function, sharing the same total arc length. b) A diagrammatic demonstration of the role of the drift coefficient (here $\mu = 32$ms) in transforming from a spatial offset of input features, to a temporal offset. In this simplified setting, $C = 0$, and there is no diffusion noise ($\sigma^2 = 0$). c) The unit square, where the axis is labelled with fractions of the maximum conduction delay $d_{\max} = 15$ms. The shaded area provides the probability that the offset of the delays satisfies some fixed value, subject to a tolerance ε. Here, the offset is $\Delta d = |d^h - d^v| = 0$ms, i.e. the special case of the probability that $d^h = d^v$, subject to some tolerance.

weight above 0) approaches that observed in Egger et al. [2020].

To understand how this can inform an estimate for a maximal Δt_{obj}, let us continue with our example of $\Delta t_T = 8$ms and $\Delta t_\perp = -8$ms. For networks using coincidence-detection, the axonal conduction delays must offset these input patterns to result in the near-simultaneous arrival of afferent spikes for one of the objects. With a distribution of delays up to a maximum of $d_{\max} = 15$ms, this will be possible, and in particular, many different pairs of delay-lines (e.g. 2.5 and 10.5 ms, or 12 and 4 ms) could support this alignment. However, if the drift coefficient was to increase, the temporal offset associated with a particular object will grow, and biologically constrained delay-lines are increasingly less likely to result in an alignment of the arriving spikes.

This can be examined mathematically by determining the probability that two randomly selected delay-lines will correct for the offset of the feature spike times of an object, subject to some tolerance ε. We will define such an event as an *alignment*. Then, given d^h and d^v, the delays associated with spikes from a horizontal and vertical bar neuron respectively, we define the magnitude of their difference as $\Delta d = |d^h - d^v|$. We subsequently define the probability of an alignment as:

$$p_{\text{alignment}} = p(|\Delta d - \Delta t_{\text{obj}}| \leq \varepsilon) \tag{3.39}$$

Note that absolute offsets are considered, as only the magnitude of the difference $d^h - d^v$ needs to approximately match Δt_{obj} for a pair of delay-lines to correct for one of the objects' spike times (either T or $\perp$).

Next, recall that the delay associated with any connection i is sampled from a uniform distribution, $d_i \sim \mathcal{U}(0, 15)$ms. The probability $p_{\text{alignment}}$ is therefore given by the joint probability density function of the two continuous random variables, d^v and d^h. Fortunately, $p_{\text{alignment}}$ has a straightforward geometric interpretation, which we demonstrate with a special case in Figure 3.14c. In particular, $p_{\text{alignment}}$ corresponds to the area of the shaded regions of the unit square.

As previously noted, $\Delta t_T \neq \Delta t_\perp$ must hold for the objects to be distinguished by a coincidence detecting neuron. As such, the possibility of receiving neurons encoding object

identity requires that $\Delta t_{\text{obj}} > 0$. Furthermore, while the delays would (from a computational viewpoint) ideally be discrete and result in perfect alignments, this is exceptionally unlikely given continuous values, and so some consideration is also given for imperfect alignments via $\varepsilon > 0$. Note, however, that additional sources of noise will exist in the system, and so "alignments" that take place with a large offset (i.e. $\varepsilon >> 0$) are unlikely to have any computational relevance, unless the receiving neuron integrates signals over a very large window, a possibility that we will return to later. Figure 3.15a demonstrates graphically how $p_{\text{alignment}}$ is influenced by both Δt_{obj} and ε. As expected, when ε is small, the probability of an alignment is exceptionally low. More importantly, as Δt_{obj} increases, the shaded region progressively vanishes, even with a large ε. These effects are shown quantitatively in Figure 3.15b, where we use a random number generator to estimate $p_{\text{alignment}}$ (i.e. sampling delays and evaluating the resulting alignments) at the relevant parameter points.

It is worth clarifying a few effects demonstrated in Figure 3.15a. With $\Delta t_{\text{obj}} > 0$, the joint probability is described by two separate regions, as the sign of $d^h - d^v$ is irrelevant for an alignment. If $\varepsilon > \Delta t_{\text{obj}}$, these separate regions will overlap (shown for example at $\Delta t_{\text{obj}} = 1\text{ms}$, $\varepsilon = 2\text{ms}$). Finally, for any given ε, the total region (i.e. $p_{\text{alignment}}$) is maximal when $\varepsilon = \Delta t_{\text{obj}}$. This is shown, for example, at $\Delta t_{\text{obj}} = 1\text{ms}$, $\varepsilon = 1\text{ms}$, and is reflected in the local maximums seen in Figure 3.15b.

The key finding in Figure 3.15b, however, is that, as Δt_{obj} grows, $p_{\text{alignment}}$ falls, even for a reasonably tolerant ε. Importantly, this is a result of the bounded conduction delays. Thus, $\Delta t_{\text{obj}} = 8\text{ms}$ (corresponding here to $\mu = 32\text{ms}$ per equation 3.38) is an appropriate value to satisfy both $\Delta t_{\text{obj}} > 0$ and $p_{\text{alignment}} > 0$. However, larger values of Δt_{obj} are less likely to have computational relevance to a model constrained by experimentally estimated conduction delays.

We emphasize that, having a larger drift coefficient and encoding smaller objects (e.g. $\mu = 64\text{ms}$, $\Delta x_T = -\Delta x_\perp = 0.125$), would still result in an appropriately constrained value of Δt_{obj} (i.e. $\Delta t_{\text{obj}} = 8\text{ms}$). However, this will not help support a larger integration window in order to enable noise robustness, as the integration window (later discussed) is ultimately bounded by Δt_{obj}. As such, our analysis suggests that the only benefit of a

higher drift coefficient than our estimate of 32ms would be to enable more extreme online translation invariance. In general however, the results from both Havenith et al. [2011] and Egger et al. [2020] have informed similar limits on the temporal offset of features that the brain could actually make use of. In our Discussion in Chapter 4, we will discuss the relevance of myelin plasticity for potentially relaxing the constraint otherwise imposed on Δt_{obj}.

3.6.2 Network Connectivity

3.6.2.1 Dense Networks

Having established upper estimates of μ and Δt_{obj}, we will next consider the influence of the connectivity of the network, and in particular the number of afferent synapses per output neuron. We first consider this under the setting of a smaller, dense network. Recall that our aim is to estimate the hyperparameter space in which the proposed computations could feasibly operate within the brain, accounting for both biological and computational constraints. In order to simplify our analysis, we define our network of interest as having the following properties:

- Connections between the input and output layers are all-to-all at the outset. Note therefore that the number of afferent synapses for each neuron in the output layer is equivalent to the size of the input layer.
- Connections have a conduction delay sampled from a uniform distribution $d_i^{exc} \sim \mathcal{U}(0, 15)$ms.
- Unless stated otherwise, $\Delta t_{\text{obj}} = \Delta_T = -\Delta_\perp = 8$ms, corresponding to the simplified spatial-configuration function considered in the previous section.
- An output neuron is considered to receive a delay-line alignment if the signals from an object would arrive within 1ms of one another (i.e. $\varepsilon = 1$ms).

In our simulations of spiking neural networks in Chapter 4, we will use a network with similar properties to this dense network, as it will allow us to efficiently evaluate the performance of the system at a diverse range of hyperparameter settings, while still maintain-

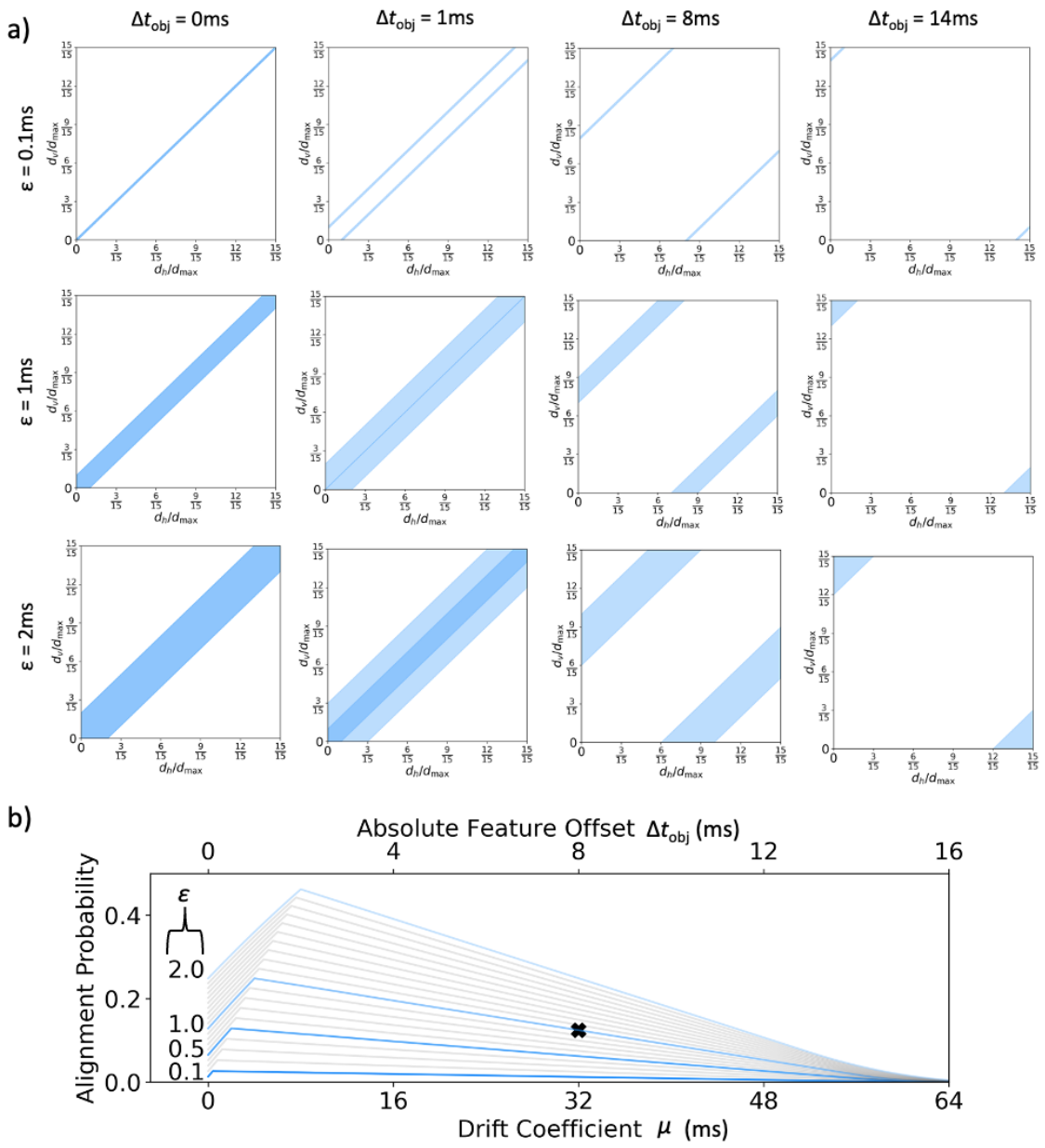


Figure 3.15: *The Interaction of* Δt_{obj} *and* ε *on* $p_{\text{alignment}}$. The subplots in (a) show graphically how the probability of an alignment occurring (the area of the shaded region) changes as Δt_{obj} and ε vary. b) Axonal conduction delays are randomly sampled from a uniform distribution $\mathcal{U}(0, 15)$ms. Shown is $p_{\text{alignment}}$ as a function of Δt_{obj}, and estimated through numerical simulation. The relationship is shown for multiple levels of ε, and the x-axis includes the corresponding value of μ given an object that composes a quarter of the receptive field (determined with equation 3.38). Function lines are highlighted in blue for $\varepsilon = 0.1$ms (i.e. virtually synchronous arrival times following the two random conduction delays), as well as 0.5, 1.0, and 2ms. The cross indicates the probability ($p_{\text{alignment}} \approx 0.124$) of an alignment with $\varepsilon = 1$ms, and given $\Delta t_{\text{obj}} = 8$ms.

ing generalizability to a more realistic, sparse setting, as discussed below. Note, however, that these simulations will not make the same simplifying assumptions about the nature of the spatial-configuration functions.

We first consider a biological basis for estimating the number of afferent connections per output neuron in our network. To do so, we highlight again the columnar organization of the brain, wherein minicolumns typically encode a common feature over a shared receptive field. These minicolumns each consist of around 200-250 neurons in the primary visual cortex [Mountcastle, 1997]. Only a proportion, perhaps around 100, of these cells will be complex cells (i.e. have invariant responses over a larger receptive field) [Gilbert, 1977]. Finally, a reasonable estimate of connection probability in the brain is 10% [Lefort et al., 2009]. Thus, a downstream neuron would receive approximately 10 afferent connections from each of two minicolumns, resulting in a total of 20 strong afferent connections determining object selectivity. As such, we model our input layer (and for simplicity, our output layer) as each containing 20 neurons with all-to-all (i.e. dense) feed-forward connectivity.

As we will now demonstrate, 20 afferent connections per output neuron corresponds to a reasonable value on a computational basis as well. We emphasize that each output neuron in our simulation also receives thousands of weaker, firing rate inputs described by Poisson statistics (Methods, Section 3.5.1). Our interest however is in sparse, stronger synapses arriving from minicolumns which may assist in distinguishing the objects through some level of spatial encoding and coincidence detection.

To understand the significance of the number of afferent synapses per neuron, we begin by highlighting that this will directly affect two key properties of the network, which are the probability of an output neuron receiving temporally aligned inputs, and the interspike interval (ISI) of afferent spikes.

For the former, recall that our interest is in whether STDP alone is sufficient for output neurons to emerge that respond to a specific object's presentation, and not another. For this to hold, it is necessary that there are output neurons which receive afferent delay-lines that correct for the temporal offset in the input pattern (resulting in the approximately coincident arrival of signals). Earlier, we described $p_{\text{alignment}}$ in the setting where only

two axonal conduction delays were being compared. In reality, each output neuron will receive synapses from multiple input assemblies. As such, so long as one of these pairs results in an alignment within the tolerance of $\varepsilon = 1$ms, then the output neuron can potentially encode an object.

Furthermore, output neurons can exist which have multiple alignments, for which these can either be for only one object, or for both. In the case of the latter, standard mechanisms of competition, such as lateral inhibition and activity sparsity, can ensure learning enables some output neurons to respond uniquely to the presence of one object.

While in theory it would be sufficient for only one output neuron with alignments to exist for each object, the network will only be viable as a biological system if it is robust to sources of noise, and so at a minimum the output layer should contain a handful of neurons with successful alignments.

Mathematically, for an output neuron receiving pairs of synapses from a certain number of assemblies (n_{assem}), the probability of it receiving exactly $n_{\text{alignment}} = k$ delay-line alignments is provided by the probability mass function of the Binomial distribution $\mathcal{B}(n_{\text{assem}}, p_{\text{alignment}})$:

$$p(n_{\text{alignment}} = k) = \binom{n_{\text{assem}}}{k} (p_{\text{alignment}})^k (1 - p_{\text{alignment}})^{(n_{\text{assem}} - k)} \tag{3.40}$$

where, given the current network assumptions (and as estimated in the previous section), $p_{\text{alignment}} \approx 0.124$. Note that we only consider alignments within cell assemblies, as it is only these neuron subsets that share any correlation in the times at which they spike.

As noted earlier, where multiple alignments take place, these may still correspond to a common object. Given the k alignments that an output neuron receives, the probability that these all offset input patterns for the same object is simply :

$$p_{\text{common:object}} = 0.5^{k-1} \tag{3.41}$$

where there is an equal probability that any alignment is encoding a particular object.

Combining these probabilities, we can determine as a function of the number of afferent synapses (i.e. $2n_{\text{assem}}$, as each assembly is associated with two input neurons) the probability that an output neuron receives at least one alignment, and that these are all for a single object, which is given by:

$$p_{\text{single:aligned}} = \sum_{k=1}^{n_{\text{assem}}} p(n_{\text{alignment}} = k) * (0.5^{k-1}) \tag{3.42}$$

This summation takes into account every possible number of alignments (k) that may take place (for $k \geq 1$), up to the maximum possible of n_{assem}. Finally, the probability that a neuron receives multiple alignments, albeit for both objects, is simply given by:

$$p_{\text{both:aligned}} = 1 - (p_{\text{single:aligned}} + p(n_{\text{alignment}} = 0)) \tag{3.43}$$

In Figure 3.16, we plot $p_{\text{single:aligned}}$ and $p_{\text{both:aligned}}$ as a function of the number of afferent synapses. It can be seen that the probability of a neuron receiving an alignment quickly exceeds 50% as the number of afferent synapses increases. As the number of afferent synapses increases further, alignments for both objects become more common than for single objects. As we will later demonstrate however, this is not observed in more plausible, sparse networks, indicating that this setting (many neurons having alignments for both objects) should be avoided in any modelling approach.

While increasing the number of afferent synapses increases the probability of output neurons with delay-line alignments, this also introduces a computational challenge. In particular, a greater number of input synapses will also result in a smaller, afferent interspike interval (ISI). This is significant, as we are concerned with neurons performing coincidence-detection. When considering integration that involves two spikes to generate an output spike (i.e. large-weight synapses), this generally requires that the ISI be larger than the integration window of a neuron [König et al., 1996]. Intuitively, neurons should integrate inputs relevant to the stimulus, but not random spikes arriving from other neurons. As we have already noted however, the system's tolerance to sources of noise such as jitter *increases* with the size of the integration window. In particular, we will later esti-

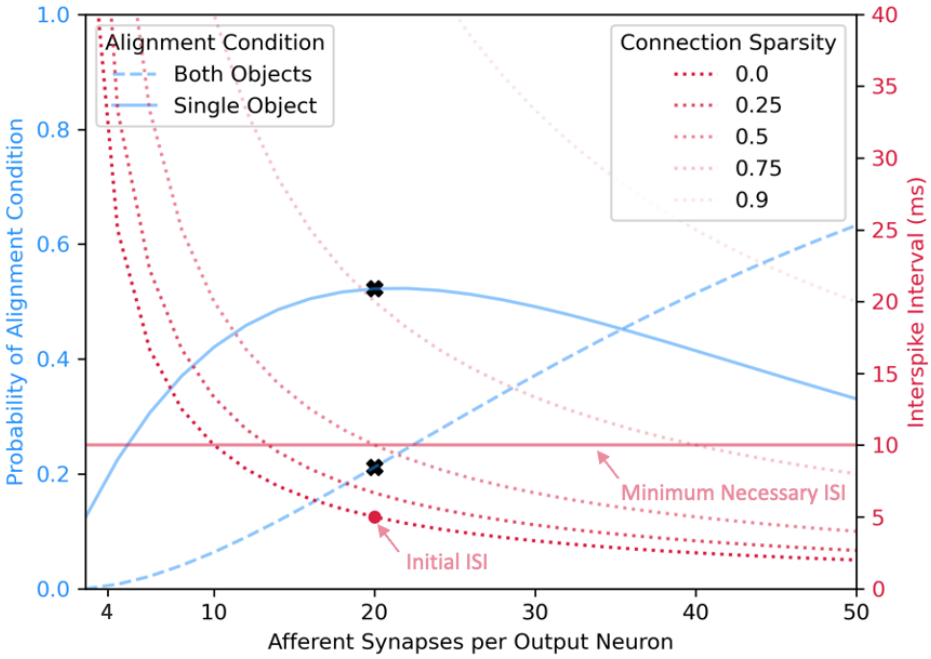


Figure 3.16: *The Influence of Afferent Synapse Numbers (Dense Connectivity).* In a dense network with all-to-all connectivity at the outset, the probability of an output neuron having a particular delay-line alignment condition is shown along the left axis as a function of the number of afferent synapses (blue). On the right axis is the afferent interspike interval as a function of the number of afferent synapses (red; given a presentation interval of 100 ms). For a network to successfully distinguish the two objects, it should have neurons with aligned delay-lines, while also maintaining a reasonably high interspike interval, indicating that the network will need to prune weights in order to achieve temporal sensitivity. The crosses indicate the probability of alignment for a single object (~ 0.52), or both objects (~ 0.21) given 20 afferent synapses. The red dot indicates the initial interspike interval, given 20 afferent synapses and no pruning (ISI=5ms).

mate a biologically plausible and computationally suitable integration window of around 8ms, implying a minimum ISI of around 10ms for robust encoding. As can be seen in Figure 3.16, the ISI decreases as the number of afferent synapses increases, presenting a challenge to achieving such a high ISI. However, as a final element, the sparsity of weights plays a counter-active role, raising the possibility that sparsity following learning might enable the necessary afferent ISI for coincidence-based computations.

In summary, this results in a trade-off, where neurons should receive a sufficient number of synapses for delay-line alignments to be likely, while also receiving sufficiently few afferent spikes to support neurons that perform coincidence detection. Our prior, biologically motivated estimate was for 20 strong afferent synapses per output neuron. Figure

3.16 shows that such a value is well suited for the first of these constraints, corresponding essentially to the maximizer (technically at 22) for single-object alignments. In particular, this results in the majority of output neurons having a delay-line alignment, of which around 50% are for a single object, and around 20% are for both. We note again that our next section will demonstrate that single-object alignment is the typical setting in the more plausible situation of larger, sparse neural networks, and therefore an element worth capturing in our model. Regarding the second constraint, 20 afferent synapses does *not* assume the necessary ISI at the outset, but following increased sparsity due to synaptic weight changes, the network may be able to achieve a sufficiently high effective ISI.

We highlight that 20 neurons in the input layer corresponds to an afferent stimulus rate of 200 Hz at the outset, and an interspike interval of 5ms. Furthermore, this ISI estimate does not account for the thousands of weaker, Poisson input spikes that we will also simulate in Chapter 4. Taken together, temporally precise computations will be a significant challenge in this dense network. The results in Chapter 4 will demonstrate that STDP provides a mechanism to naturally promote sparsity, enabling the network to converge to a much higher effective ISI, and thereby establish sensitivity to coincident inputs.

3.6.2.2 Sparse Networks

A natural concern is how this evaluation relates to more plausible networks of neurons, which are both larger, and have sparse connectivity. To assess this, we continue with our motivating setting of two cortical minicolumns as our input, with biologically plausible connection sparsity. Rather than 20 neurons, the input layer thus consists of 200 neurons (100 for each feature) that send feed-forward projections that synapse with a probability of 0.1. Excluding these differences, the network follows the same assumptions as the dense, 20-neuron network, such as the distribution of conduction delays, and the definition of an alignment of delay-lines. Each output neuron will still receive around 20 strong connections (in addition to thousands of weaker background inputs), and thus the same evaluation regarding the afferent ISI holds as for the dense network above. The question for the following section is therefore to determine whether, in a sparse network, a reasonable number of output neurons with delay-line alignments will still emerge.

For a given output neuron to be able to learn the object encoded by an input assembly, it must i) receive synapses from both neurons in the input assembly, and ii) the delay-lines should offset the stimulus' temporal pattern. In dense networks, there is all-to-all connectivity, so (i) automatically holds. In a sparse network however, there is no guarantee that an output neuron will receive a synapse from both neurons composing an assembly.

The Probability of a Match To explore this further, we begin by defining $n_{a:\text{size}}$, the number of features in each assembly. Up until now, we have generally considered the case $n_{a:\text{size}} = 2$. We subsequently define the event which we will call a "match"; this is the event that, when sampling projections to an output neuron, we sample from a particular assembly feature neuron, as well as all the other feature neurons composing that assembly. We note that any given output neuron might end up with matches with multiple input assemblies, quantified with m_{count}. To determine the probability of a match, we note that p_{syn} is the overall probability of a synapse forming between any given input and output neuron, while $p_{\text{syn}|a}$ is the conditional probability of a synapse from assembly a on an output neuron, given that the output neuron has already received a synapse from another neuron in assembly a. Under pure chance, this is simply the overall sparsity of connections in the network, i.e. $p_{\text{syn}|a} = p_{\text{syn}} = 0.1$. We will soon comment on the conditions under which we might observe $p_{\text{syn}|a} > p_{\text{syn}}$. When sampling the connectivity to an output neuron, the probability that we sample from the first feature associated with a particular neural assembly, and subsequently sample from the other $(n_{a:\text{size}} - 1)$ feature neurons of that assembly, is then given by:

$$p_{\text{match}} = p_{\text{syn}} * (p_{\text{syn}|a})^{(n_{a:\text{size}}-1)} \tag{3.44}$$

Note that, with $n_{a:\text{size}} > 2$, we make the simplifying assumption that $p_{\text{syn}|a}$ does not change with each subsequent synapse received from an assembly, although in reality, this conditional probability would likely progressively increase.

Recalling that n_{assem} is the number of input assemblies, the probability of an output neuron having any given number of matches is then described by the probability mass func-

tion of the Binomial distribution $\mathcal{B}(n_{\text{assem}}, p_{\text{match}})$:

$$p(m_{\text{count}} = x) = \binom{n_{\text{assem}}}{x} (p_{\text{match}})^x (1 - p_{\text{match}})^{(n_{\text{assem}} - x)} \tag{3.45}$$

as each assembly can be considered independently, and may or may not send any projections to the output neuron (captured by p_{match}). Note how as p_{match} approaches 1, $p(m_{\text{count}} = n_{\text{assem}})$ will also converge to 1, i.e. the maximum possible number of assemblies. This is the condition of all-to-all connectivity.

The probability mass function can then be used to determine the probability that any given output neuron has at least one match. In particular, this is simply:

$$p(m_{\text{count}} \geq 1) = 1 - (1 - p_{\text{match}})^{n_{\text{assem}}} \tag{3.46}$$

where the second term is the probability that $p(m_{\text{count}} = 0)$. In the event that an object is composed of more than two features encoded via spatial-configuration functions, n_{assem} should be unaffected (e.g. as additional input minicolumn populations are considered). However, it is clear that as $n_{a:\text{size}}$ grows, p_{match} (and therefore $p(m_{\text{count}} \geq 1)$) can decrease significantly. At the same time, $p_{\text{syn}|a} = p_{\text{syn}}$ assumes that the connectivity in the network is entirely random, at odds with experimental findings [Lefort et al., 2009, Takahashi et al., 2010]. Figure 3.17a shows the probability of at least one match on an output neuron as a result of both increasing $n_{a:\text{size}}$ and $p_{\text{syn}|a}$. As can be seen, these two counter-act one another, such that under diverse conditions, a reasonable probability of matches (e.g. $p((m_{\text{count}} \geq 1) \geq 0.5)$ is observed.

A variety of anatomical and plasticity factors are likely to contribute to feed-forward projections that are correlated amongst feature neurons in the same assembly, resulting in $p_{\text{syn}|a} > p_{\text{syn}}$. For example, the lateral projections of neurons in cortical columns are often highly structured depending on the depth of the neuron. The probability distributions of these projections can be concentrated at a resolution narrower than a single cortical layer [Lefort et al., 2009], which could help form structured assemblies that include the output (i.e. downstream) neurons. Consistent with this, *in vitro* measures of neural assemblies

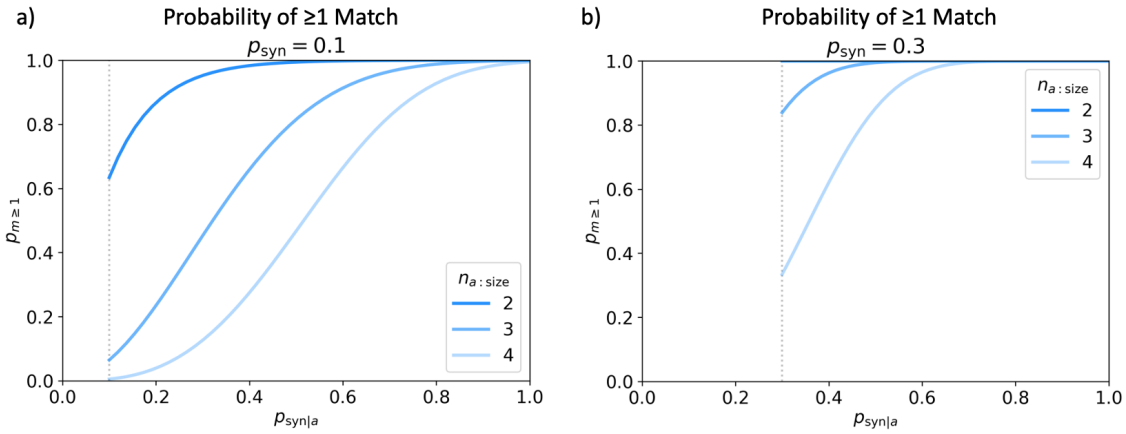


Figure 3.17: *The Influence of Correlated Projections and the Size of Neural Assemblies.* Shown is the probability of an output neuron having at least one match as a function of both varying the degree of correlation in the assembly projections, as well as the number of neurons composing a given assembly. Results are provided for both a setting of high initial sparsity (a), as well as lower initial sparsity (b), as might be found earlier in development. Note that reducing the initial sparsity affects *both* p_{syn} and $p_{\text{syn}|a}$. As such, even when $p_{\text{syn}} = p_{\text{syn}|a}$ (i.e. connectivity is random), the probability of matches increases significantly. In this particular setting, the function line for $n_{a:\text{size}} = 2$ is difficult to perceive as it is at 1.0 throughout.

that share temporally structured activity have shown that participating neurons are significantly more likely to project to a common downstream neuron than neurons with no temporal association [Takahashi et al., 2010].

It may also be that the baseline synaptic probability is higher earlier in development (e.g. $p_{\text{syn}} = 0.3$). Indeed, humans go through a stage of a rapid increase and then progressive decrease in synaptic density during development [Huttenlocher and Dabholkar, 1997]. Such a modification shifts the probability matches significantly (Figure 3.17b). Unlike correlated projections however, increasing p_{syn} will increase the number of synapses, which may introduce computational complications by narrowing the (initial) afferent interspike interval. This might therefore necessitate additional learning mechanisms to promote an early, coarse form of assembly-based synaptic pruning.

The Probability of Alignments Having established the probability of matches in sparser networks, we note again that forming a match is only the first condition of a delay-line alignment. Recall that our aim is to determine whether output neurons with alignments could still emerge in sparse networks, without requiring a prohibitive increase in the num-

ber of afferent synapses. To examine this, we can adapt equation 3.40 from dense networks, where the number of afferent assemblies is now determined by the number of matches, i.e.:

$$p(n_{\text{alignment}} = k) = \binom{m_{\text{count}}}{k} (p_{\text{alignment}})^k (1 - p_{\text{alignment}})^{(m_{\text{count}} - k)} \tag{3.47}$$

Together with the probability mass function of the number of matches (equation 3.45), these equations provide us with the probability of an output neuron having alignments for a single object in sparse networks:

$$p_{\text{single:aligned}} = \sum_{x=1}^{n_{\text{assem}}} p(m_{\text{count}} = x) * \left(\sum_{k=1}^{m_{\text{count}}} p(n_{\text{alignment}} = k) * (0.5^{k-1}) \right) \tag{3.48}$$

Note that in the setting of all-to-all connectivity, $p(m_{\text{count}} = n_{\text{assem}}) = 1$, and $p(m_{\text{count}} = x) = 0$ for all other $x \neq n_{\text{assem}}$, reducing equation 3.48 to equation 3.40.

The probability that no alignments take place is as follows:

$$p_{\text{none:aligned}} = \sum_{x=0}^{n_{\text{assem}}} p(m_{\text{count}} = x) * p(n_{\text{alignment}} = 0) \tag{3.49}$$

Note the change in the sums, and that it is straightforward to use equations 3.48 and 3.49 to determine the probability that an alignment takes place for both objects.

To connect this equation to our earlier analysis of the drift coefficient, we consider the probability of an output neuron in a sparse network receiving an alignment as a result of various drift coefficients (μ), where the object takes up a quarter of the receptive field, and the maximal conduction delay is 15ms. These results are shown in Figure 3.18, where we once again also vary both p_{syn}, $p_{\text{syn}|a}$, and $n_{a:\text{size}}$. These results demonstrate the significance we earlier outlined for the magnitude of the drift coefficient. In particular, while larger drift coefficients are useful for robustly encoding objects described by the unique relations of features, these can be problematic for the probability of alignments

taking place, which is particularly acute in sparse networks.

We highlight that, although the probability of alignments is of course lower than in the dense setting, we are also now operating with much larger networks. For example, the expected number of output neurons with a single-object alignment is simply $n_{\text{output}} * p_{\text{single:aligned}}$. Thus, with an output layer of 100 neurons, random projections ($p_{\text{syn}} = p_{\text{syn}|a} = 0.1$), $n_{a:\text{size}} = 2$, and a drift coefficient of $\mu = 32$ms, we will have approximately $n_{\text{output}} * p_{\text{single:aligned}} = 100 * 0.113 \approx 11.3$ neurons with single-object alignments. In our dense network with 20 input and output neurons, we observe $n_{\text{output}} * p_{\text{single:aligned}} = 20 * 0.522 \approx 10.4$. Given the thousands of synapses that neurons project, as well as the quantity of neurons dedicated to primate vision [Felleman and Van Essen, 1991], there would likely be many alignments even with a low associated probability.

We emphasize with reference to Figure 3.18 that the probability of an alignment for *both* objects is minimal, so long as the degree of correlations remains moderately low. This is a result of the requirement that not only must multiple matches exist, but that these must subsequently result in multiple alignments, and in addition the alignments must be for different objects. For example, whereas the number of single-aligned neurons in both the sparse and dense network are very similar (around 10), $n_{\text{output}} * p_{\text{both:aligned}} = 100 * 0.004 \approx 0.4$ neurons for the sparse network, in contrast to $n_{\text{output}} * p_{\text{both:aligned}} = 20 * 0.212 \approx 4.2$ neurons for the dense network. These findings support the choice of parameters that will be used later in our dense spiking neural network simulations, as the network with 20 input and output neurons aligns well with properties observed in larger sparse networks (i.e. both the quantity of output neurons with single-object alignments, and the relatively minor role of neurons with both-object alignments).

As $n_{a:\text{size}}$ increases, Figure 3.18 demonstrates that correlated projections become increasingly important for the viability of alignments taking place. We first note that these results are concerned with the setting where all of the neurons in an assembly successfully align their synapses. It is possible in the setting where $n_{a:\text{size}} > 2$ that only a subset of neurons will result in an alignment, yet in such a setting these feature neurons could still have a computational role in modifying the response profile of a neuron. In other words, given an assembly of neurons with temporally associated activity, some of the output neurons they

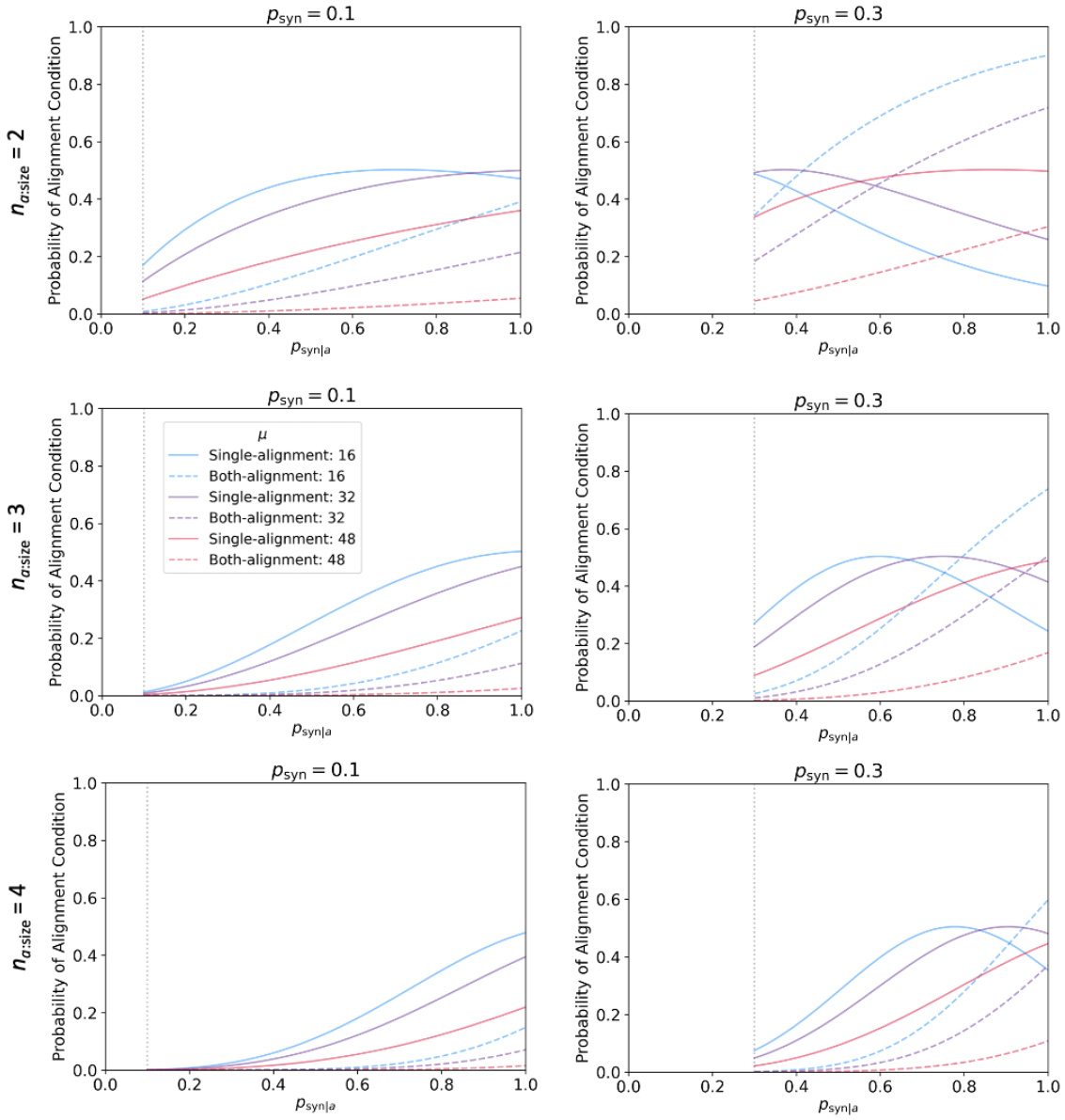


Figure 3.18: *The Probability of Alignments in Sparse Networks.* Shown is the probability of an output neuron having an either single-object or both-object alignment condition in sparse networks. This probability is provided under the setting of varying p_{syn} (the separate columns), and $n_{a:\text{size}}$ (the separate rows). Within each plot, results for single (solid line) and both (dashed line) alignment conditions are provided, given a drift coefficient μ of 16 (blue), 32 (purple), and 48 (red), and as $p_{\text{syn}|a}$ is varied.

project to may receive alignments for all of the neurons, and some output neurons may only receive alignments for a subset, but in both conditions, the projecting assembly could alter the response profile of the output neuron based on the spatial offsets of features.

We also emphasize that output neurons which don't receive any alignment can still use the information they receive in the form of spikes. In particular, this will still represent a signal about the presence of a feature. Absent an afferent alignment of other neurons *from that particular assembly*, receiving neurons will simply not have access to additional spatial information. In this way, such neurons degenerate to the outputs of a pooling operation, which as we have noted, can be capable in their own right. Thus, the output population which does not receive temporal alignments is not wasted, a key point given the relatively low probability of alignments.

Overall, however, the above results for both the probability of matches and alignments emphasize the challenges that emerge in appropriately routing network signals when faced with the reality of sparse connectivity. This is a frequently encountered challenge in computational theories of the brain, and as such even biologically motivated models of cortical processing frequently assume all-to-all or otherwise highly structured connectivity (e.g. Manohar et al. [2019], Ahmad et al. [2020], Pogodin et al. [2021]).

These results are also consistent with our earlier discussion in Section 3.3 of the challenges that emerge when considering higher-dimensional feature-sets if these are strictly held to all form spatial-configuration functions. Once again, rate-coded neurons projecting weaker synapses could help relax the requirement for correlated projections. In particular, these neurons can project more widely or with greater redundancy, as there is no requirement that they arise from the same input "assembly" (i.e. set of feature neurons spiking with a specific temporal structure). It is sufficient in that case for the output neuron to receive a synapse from any one of the rate-based neurons encoding a feature of interest, thereby ensuring that $n_{a:\text{size}}$ remains small.

Summary of Sparse Connectivity Bringing this analysis together, plausible conditions exist under which a sufficient number of neurons would receive delay-line alignments that spatial-configuration functions could play a computational role in the brain. Such alignments can emerge in sparse networks without resulting in an inappropriately high number

of strong synapses arriving at each neuron, and thus without resulting in a dramatic fall in the afferent ISI. Our specific findings also support the validity of the efficient simulations we perform in Chapter 4 with smaller, dense networks.

Importantly, however, a significant proportion of the network will not receive aligned inputs. As we have noted, these output neurons are not wasted (they can still encode information in a typical, less spatially-sensitive manner), however this finding suggests it is unlikely that the spatial-configuration functions as presented would be the central basis of visual processing in the brain. Rather if it is found in biology, it is likely to represent a more specialized computation. In our Discussion in Chapter 4 (Section 4.6), we will address the potential role that myelin plasticity could play in enabling the computation to be more universal. We will also discuss experimental evidence that the robust computations we discuss may have an outsize role for particularly familiar stimuli such as faces and words, and how this might be reconciled with the notion that these computations could be useful for online translation invariance with novel objects.

3.6.3 The Integration Window

With an established upper bound on both μ and Δt_{obj}, as well as an understanding of the influence of the network connectivity, we will now examine the significance of the integration window of a neuron. We will then bring these considerations together. This unification of concepts will shed light on the bounded parameter space they create wherein the proposed computations could potentially take place within the brain.

The integration window of a neuron can be variously defined as the temporal interval over which synaptic potentials undergo summation that is significant [Theunissen and Miller, 1995], "effective" [König et al., 1996, Gabernet et al., 2005, Léger et al., 2005], or non-linear [Usrey et al., 2000, Cardin et al., 2010], or alternatively, the interval within which synaptic inputs must summate in order to reach the spiking threshold [Pouille and Scanziani, 2001, Lücken et al., 2017]. We are interested in exploring neurons that communicate information by emitting an output spike, and so we will work with the latter definition. Depending on the precise definition used, as well as whether neurons are considered within the context of influences such as feed-forward inhibition and changing

network activity levels, estimates of the integration window vary. Examples of experimentally derived estimates range from 2-5ms [Pouille and Scanziani, 2001], 7ms [Usrey et al., 2000], 1-10ms [Gabernet et al., 2005], 8-16ms [Cardin et al., 2010], and 10-20ms [Léger et al., 2005]. In the following, we will consider the biological and computational constraints of our network of interest. These considerations will suggest that a value in the range of 8ms would enable optimal robustness, although in our simulations, we will also consider the effects of both smaller (4ms) and larger (12ms; 16ms) values.

We note that the integration window is not necessarily equivalent to the time constant of a neuron's membrane potential, which is generally estimated for cortical neurons to be in the region of 8-20ms [Shadlen and Newsome, 1994]. Indeed, the integration window can be much smaller, due to factors such as feed-forward inhibition [Pouille and Scanziani, 2001, Gabernet et al., 2005], although occasionally no distinction between the two is made [König et al., 1996]. This is unsurprising, as a neuron with a smaller membrane time constant will naturally integrate afferent spikes over a narrower window, and thus these are related properties. In our simulations, we use neurons with conductance-based synapses. While these have the advantage of biological accuracy, deriving an accurate estimate of the effective membrane time constant is not trivial [Rossant et al., 2011]. For these reasons, we once again emphasize the definition of the integration window that we will use, which is the interval over which a pair of feature spikes must arrive in order to reach the spiking threshold. We also highlight that neurons can have membrane time constants and integrative properties on much longer time scales than those that we consider [Koch et al., 1996]. For example, specialized neurons in the fruit fly have membrane time constants on the order of 100s of ms, enabling them to integrate and thereby evaluate noisy information about the concentration of an odorous gas in two different directions [Groschner et al., 2018]. Such temporally broad, integrative computations are likely to be important for decision making under ambiguous, noisy conditions [Bogacz et al., 2006], although this more noisy setting with longer time constants [Koch et al., 1996] is not one that we consider in this work.

We now examine the significance of a neuron's integration window. When neurons are viewed as temporal coincidence detectors, emphasis is typically directed to the narrow-

ness of the integration window, as this ensures that neurons are sensitive only to highly aligned temporal inputs [König et al., 1996]. Our prior analyses showed that various network properties tend to result in a computational trade-off, and once again, this will be the case for the integration window.

For a given output neuron to encode information about the presence of an object, it must spike in response to the coincident arrival of signals from the constituent features. While these features would ideally arrive with perfect temporal alignment, this will not occur in biology for a variety of reasons. The most straightforward of these is noise in the form of jitter, for which a wider integration window would be desirable [Masquelier, 2018]. In addition, where a good alignment takes place at one location in the position function, diffusion noise will degrade this alignment at other locations (i.e. 'structural' noise). Finally, axonal conduction delays are unlikely to result in perfect temporal alignment at a receiving neuron, compounding the noise introduced by jitter and diffusion in the position function. For these reasons, a more lenient integration window, such that coincident arrival is tolerated on the order of tens of ms, would enable greater robustness.

The importance which is normally given to a narrow integration window is not without reason however. If the integration window is too broad, then random, afferent spikes from neurons that do not form part of the input stimulus or neural assembly of interest will also be integrated, leading to the destruction of temporal-based information. As such, the integration window should, as a rule of thumb, be less than the ISI, although a precise choice of the integration window is not straightforward to derive [König et al., 1996].

In addition to this limitation of using a wider integration window, the values of μ and Δt_{obj} that we earlier estimated impose an additional constraint on using large integration windows. In particular, for an output neuron to distinguish an object, any given pair of delay-lines would ideally result in alignment with one object over another. In the noiseless example we have previously considered with a drift coefficient of $\mu = 32$, only a fraction of delay-line pairs will result in the necessary offset of ± 8ms, whereas delay-line pairs with approximately *equal* delays will be more common. For example, with $\Delta t_{\text{obj}} = 8$ms and $\varepsilon = 1$ms, the probability that any two delay-lines sampled from $\sim \mathcal{U}(0, 15)$ms will result in an alignment is approximately 0.12. The equivalent probability given $\Delta t_{\text{obj}} =$

1ms is approximately 0.25. This same principle was demonstrated in Figure 3.14.c, where smaller drift coefficients were associated with a higher overall probability of delay-line alignments.

What is the significance of this for the integration window? If we continue to imagine stimuli that are defined by temporal offsets of $\Delta t_{\text{obj}} = \Delta_T = 8$ms, the above probabilities demonstrate that many of these signals will travel via approximately equal delay-lines, *arriving* at the receiving neuron with an offset of around 8ms. While a neuron with a narrow integration window would not respond to such temporally misaligned inputs, a neuron with a broader integration window will. This is important, as regardless of the stimulus presented, these delay-lines will result in the same offset of the signals when they arrive, driving a broadly-integrating neuron to fire. The consequence is that the output neuron will lose specificity to a particular spatial arrangement of features, and will respond indiscriminately to either stimulus (i.e. from its point of view, $\Delta_T = \Delta_\perp$). Note that not all neurons will receive these non-specific delay-lines, and sources of competition (such as lateral inhibition) could help neurons to prune them. For this reason, this does not introduce a hard constraint on the integration window. However, the ability to distinguish spatiotemporal patterns represents an additional benefit of a narrow integration window.

To summarise, as well as connect this discussion to the opening problem of robust object recognition, we note that the integration window plays an important role in the trade-off between invariance and form specificity. A wide integration window enables tolerance across multiple locations, even given a highly non-linear spatial-configuration function, but at the risk of being unable to distinguish different objects. An extremely narrow integration window may only respond to a stimulus at a particular location, ensuring specificity in its response, but fail to generalize to novel locations. In addition to these computational considerations, however, the integration window has practical significance at the implementation level. In particular, the integration window influences how, what is ultimately a biological system, handles other sources of noise such as jitter and spikes that arrive from irrelevant neurons.

We can now examine these aspects of the integration window together. To do so, we will visualize the integration window of a leaky integrate and fire neuron. As noted, we are

interested in the interval over which two spikes must arrive in order to drive an output spike. To visualize this, we simulate the neuron receiving spikes at different temporal offsets, assuming both spikes are associated with a maximal synaptic weight (Figure 3.19).

In our network simulations in Chapter 4, we will wish to observe the effects of different integration windows, a possibility which is demonstrated in Figure 3.19. To modify the integration window, we do not attempt to directly change the membrane time constant, which as earlier noted, is difficult to accurately estimate in neurons with conductance-based synapses. Instead, we vary the maximum possible weight in our model. Taking this approach also ensures that not all neurons are assumed to have a particular integrative property at the outset, as only a small fraction of neurons will be initialized with a weight near the maximum possible. The effects of the synaptic weight, and the resulting integration window, are shown in Figure 3.19.a. In particular, this figure shows the results of simulating a single integrate-and-fire output neuron receiving two afferent spikes, where we vary the time separating these spikes (bottom x-axis), over the course of a simulation (top x-axis). In between each delivery of an input spike pair, sufficient time (around 100ms) passes such that there is no influence across spike pairs. The different sub-plots correspond to different values of the weights associated with the afferent spikes. Finally, we note that the resting and reset membrane potentials of the output neuron are, for the purpose of this simulation, set to the approximate mean membrane potential imposed by our background input.

We also demonstrate the relationship of the integration window to factors that impact robustness, such as jitter and the interspike interval (Figure 3.19.b). We highlight that, even with simultaneous spikes, the output is always a single spike, consistent with a binary (rather than rate) information code.

Figure 3.19.b demonstrates several factors that the integration window must account for. On the one hand, robustness requires that the integration window be wide enough to result in a consistent output spike in spite of jitter. An integration window of 4ms captures a significant proportion of the misaligned spikes that result from 2ms of jitter, but this does not take into consideration other sources of noise, such as diffusion in the position functions, or sub-optimal delay-line alignments. As a result, a larger integration window is likely

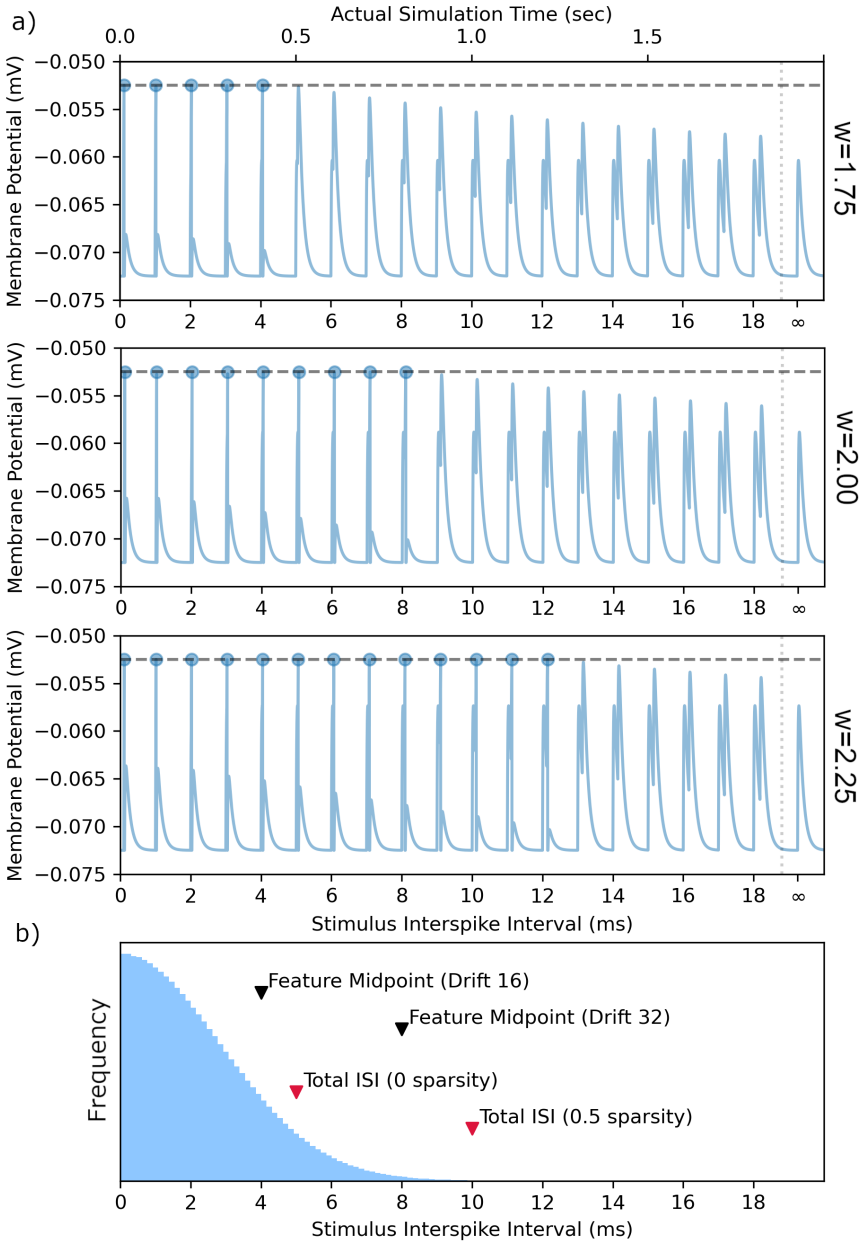


Figure 3.19: *Synaptic Weight, the Integration Window, and Robustness.* a) The output response of a neuron receiving temporally offset spikes, where the *stimulus* interspike interval is provided along the x-axis. For the different subplots, we vary the weight of the input synapses, resulting in a larger integration window as the weight is increased. The final output response (right) shows the result of an individual input spike. b) The distribution of offsets of two spikes that are initially simultaneous, but which are then each perturbed by a jitter of 2ms. Also shown are markers indicating important temporal intervals for the system. These are the Total ISI (i.e. from all 20 afferent synapses in a network), and the 'feature midpoint' (the halfway point between the spatiotemporal patterns of the two objects, assuming idealized position functions).

desirable, such as 8 or 12ms. At the same time, for neurons to be capable of distinguishing different objects (i.e. spatiotemporal patterns), the integration window should not be much larger than 8ms given a drift coefficient of 32ms, or 4ms given a drift coefficient of 16ms. Note that we can therefore predict that the most robust network will be one with the steeper drift coefficient of 32ms in combination with a broader integration window (8ms). Finally, the ISI imposes additional pressure on maintaining a smaller integration window, although this is alleviated if the network can prune connections and achieve a higher degree of weight sparsity. How much sparsity will actually emerge in the network, however, is difficult to establish a priori.

The concepts that we explore are summarised in Figure 3.20. We emphasize again that none of these constraints are hard, as they do not induce discrete, all-or-nothing effects, and might be relaxed given appropriate assumptions. Furthermore, the effects of STDP in modifying the network's properties make direct estimates of the bounded region challenging. The existence of at least two factors on each side suggests, however, that the overall space is unlikely to change dramatically. As such, the interplay of these factors indicates that the network will perform optimally when it balances tolerant integration with sensitivity to temporal coincidence detection. Encouragingly, the proposed region aligns well with experimentally derived estimates of the integration window of neurons. Furthermore, many of these estimates have a range as the integration window can dynamically change, due to factors such as inhibition and the level of general network activity. As such, it may be that neurons can dynamically adjust the interval over which they integrate spikes in order to perform optimally in a given setting. In our simulations, we will explore the extent of this region that supports robust information encoding. In particular, we will assess a 4-dimensional space determined by the jitter, diffusion noise, drift magnitude, and integration window size, framing our results within the biological plausibility of the system.

As we noted earlier, the integration window and the membrane time constant of a neuron are not typically equivalent. It is worth, however, highlighting the work of Masquelier [2018], a computational study that explored the optimal *membrane time constant* for a network performing coincidence-detection. This estimate took into account both the fre-

quency of input firing rates, and the amount of jitter in a spatiotemporal pattern. While their problem differed in several ways, such as the focus on distinguishing a spatiotemporal pattern from noise (rather than distinguishing different spatiotemporal patterns), and the absence of considerations such as diffusion noise or conduction delays, their results are relevant. In particular, they found that a time constant on the order of, at most tens of ms, provides an optimal balance between robustness to jitter, and temporal specificity in the face of high-frequency inputs. More specifically, they found that a time constant of $\approx$ 10ms corresponded to the optimal value given biologically reasonable firing rates of 1-10Hz, and jitter values of 1-10ms. We highlight that these membrane time constant values are similar to those that have been experimentally estimated [Shadlen and Newsome, 1994], and are broadly consistent with the integration window values that we will explore.

Finally, it is worth making a distinction between the integration window, and the alignment tolerance we used earlier to estimate the probability of useful delay-lines existing under various network conditions (e.g. Figure 3.14c). Unlike the integration window, the alignment tolerance is simply a threshold that is useful for estimating the functional significance of network connectivity under various settings. In general, we would expect that the alignment tolerance will need to be very narrow, as this represents the offset of the arriving signals in a *best-case* scenario (i.e. assuming no sources of noise). Hence, when visualizing the connectivity of the networks, we consider a successful alignment as falling in the range of $\pm$1ms. Delay-line alignments falling significantly outside of this level of tolerance are unlikely to have much significance once sources of noise are introduced into the network. In contrast, the integration window is an actual property of each neuron that determines its tolerance to incoming coincident signals, and therefore its responses given the arrival of spikes. We note that the integration window must be at least as large as the alignment tolerance for an alignment to have any computational relevance.

3.7 Discussion

In this chapter, we have established a framework for understanding how the relative times with which spiking neurons fire could serve as a basis for object recognition, building

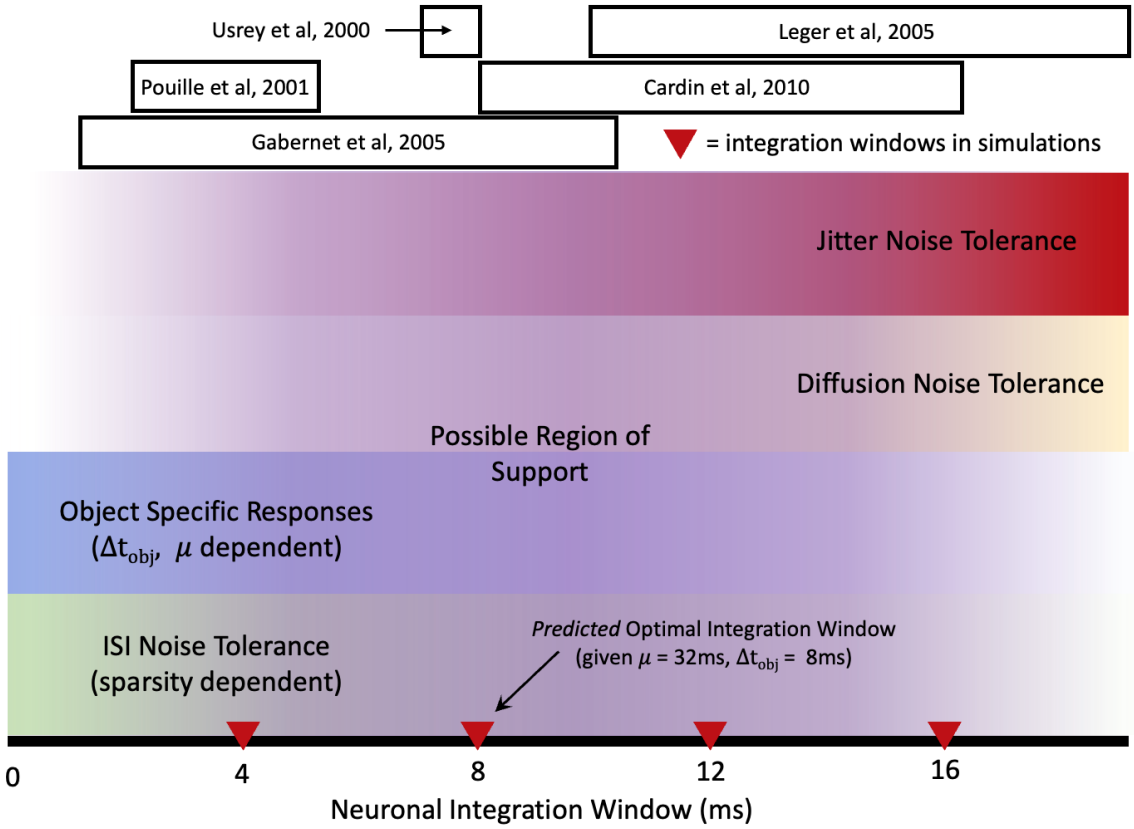


Figure 3.20: *Visualising the Space of Integration Window Constraints.* Shown is a diagrammatic representation of the compatibility of different integration window values with the computational and biological factors that have been discussed. The opaque coloured regions indicate areas where an integration window of that value would be well suited to a particular constraint; as none of the discussed constraints introduce hard boundaries, their effects are represented with a gradient. For example, 'jitter noise tolerance' is darker on the right, where larger integration windows provide more robustness to this form of noise. The purple region in the centre provides a sense of where the integration window might simultaneously satisfy all of the requirements. We highlight that the influence of the object-specific responses and ISI noise tolerance are respectively dependent on μ and Δt_{obj}, and the sparsity of weights following learning. The inset boxes provide estimates of the integration window from the respective experimental papers (a distribution that we largely cover with our simulations), although we note that none of these papers provide a perfectly equivalent means of measuring the integration window.

on ideas proposed in prior work [Davis, 2010b, Sabour et al., 2017]. In particular, we have considered the significance of spatial-configuration functions for object recognition that is robust to online translations, while also being sensitive to the spatial arrangement of a common set of features. The framework we laid out enables exploring how well such recognition performs as the system diverges from optimal properties, such as linear position functions.

We subsequently discussed the possible experimental basis of spatial-configuration functions, highlighting both the mechanisms by which firing rate codes might be transformed into a temporal code, as well as the potential role of travelling waves of electrical activity. We note again that we do not ourselves model how such encoding might actually emerge in a self-organizing and biologically plausible manner, and this remains an interesting (and important) question for future research.

Finally, we carried out an analysis of the hyperparameter space within which the proposed computations could actually take place in the brain. Our approach took into consideration both computational (would the system actually solve the task) and biological (is it consistent with experimental data) considerations. This analysis revealed a bounded space within which the proposed networks would likely be capable of robust object recognition, while also remaining within the limits of what is currently known about neurobiology. Importantly, this analysis is not static; as new experimental data emerges, this analysis can be re-visited in order to either relax, or narrow, the boundaries that we have established. In carrying out this work, we have aimed to provide a mechanistic understanding of the influence of each hyperparameter, even if the highly complex and non-linear nature of spiking neural networks resists an analytical description of the entire space, or the establishment of hard bounds.

In our next chapter, we turn to simulating such networks, which will enable us to directly test some of the predictions established here, and gain greater insight into the settings under which spatial-configuration functions could and could not underlie robust vision in the brain. The use of such simulations is important; as noted, neural networks (particularly given spiking dynamics) are highly non-linear systems, and as such, these will serve as an important adjunct to this chapter's results. As these simulations will enable a broad

discussion of both the *in silico* results and the work considered here, we reserve our full discussion to the next chapter. At that time, we will address the significance of the work, the limitations in our assumptions, and future extensions of the presented ideas.

4 | Translation Invariance in Simulated Spiking Neural Networks

4.1 Introduction

In the previous chapter, we introduced a framework for understanding how the relative times with which feature neurons spike could serve as a basis for enabling object recognition that is both robust to online translations, and form specific. This framework extended prior ideas in the literature [Davis, 2010b] to a more general setting (i.e. position functions of arbitrary form), and was followed by an analysis to establish the parameter space within which biologically plausible spiking neural networks might be able to achieve invariant representations based on such spatiotemporal patterns. In this chapter, we will carry out simulations in spiking neural networks to directly evaluate this parameter space, finding that such computations are indeed within the realm of biological plausibility, although they can also be sensitive to certain hyperparameters (in particular the drift coefficient). We emphasize that our approach is designed so as to not simply demonstrate that the proposed computations can take place at one point in parameter space, but rather to evaluate the limits of the parameter regions, and in particular the conditions under which representations lose robustness.

We begin with a discussion of related literature on the evidence of delay-line computations in biology, as well as previous simulation studies of object recognition in spiking neural networks, with a particular emphasis on the latter where they invoke heterogeneous conduction delays.

4.2 Related Literature

4.2.1 Temporal Computations With Delay-Lines in Biology

As we noted in our main introduction, the proposal that the brain can perform computations that depend on temporally precise spike times continues to divide the neuroscience

community [Baker and Lemon, 2000, London et al., 2010, Luczak et al., 2015, Williams et al., 2020]. Evidence exists for temporally precise spike times in vitro [Berry et al., 1997], as well as in early visual processing relative to stimulus onset [Reinagel and Reid, 2000, Kumbhani et al., 2007], demonstrating that neurons are in principle, capable of high precision. While these spike times become increasingly unreliable as the hierarchy of processing is ascended [Kara et al., 2000], new statistical methods are enabling the retrieval of temporally precise activity even in deeper layers of processing, where this precision need not be time-locked to points such as stimulus onset or behaviour [Havenith et al., 2011, Williams et al., 2020]. The reliability, and therefore significance, of relative neuron spike times therefore remains an open question.

Similarly, the importance of computations based on precise axonal conduction delays (i.e. delay-lines) remains unclear. In the classic model of Jeffress [1948], it was proposed that intra-aural time differences were processed by delay-lines so as to create a map of sound location. Similarly, in Reichardt [1961], delay-lines were proposed to play a role in insect motion processing. Since then, experimental evidence has been found of such delay-line computations for sound localization in barn owls [Carr and Konishi, 1988], echolocation in bats [Kuwabara and Suga, 1993], and motion detection in flies [Reichardt and Egelhaaf, 1988]. Related delay-lines models have been found to be consistent with human perception of motion [van Santen and Sperling, 1984], and the single neural responses of guinea pigs to intra-aural time differences [Shackleton et al., 2003]. More recently, Egger et al. [2020] measured the neural activity of zebra finches performing courtship singing. They observed that this neural data was best matched with spiking neural network models containing heterogeneous axonal conduction delays. Consistent with a potential computational role for delay-lines in neural processing, they measured axonal conduction delays of up to 20ms in zebra finch HVC neurons, and 15ms in rat intracortical, somatosensory neurons.

While the above examples provide evidence that neurons are capable of temporally precise computations reliant on delay-lines, it not clear to what degree such computations are restricted to specialised sensory or motor processing, rather than a more universal phenomenon. For example, courtship singing in zebra finches is a specialized motor be-

haviour that requires consistently replicating a temporally precise sequence of vocalizations. It might therefore represent an anomaly in neural processing, where heterogeneous conduction delays are uniquely beneficial.

In summary, the current evidence indicates that biological neurons are, at a physiological level, very much capable of temporarily precise computations, including those relying on delay-lines. Whether such computations are actually widely utilized in the primate brain (and for visual processing in particular), or instead represent highly specialized neural processing restricted to certain species and modalities, remains to be determined. In this chapter, we will use spiking neural network simulations to determine the settings under which such computations solve the task of robust object recognition (if at all), and the conditions under which this fails. These findings will help us to establish the degree to which these computations could be plausibly found in the primate visual cortex.

4.2.2 Spiking Neural Networks : Delay-Lines and Object Recognition

A wide literature exists on modelling the potential significance of heterogeneous delay-lines for various computations. In addition to the classic models of Jeffress [1948] and Reichardt [1961] noted above, the potential relevance of conduction delays for cortical representations was recognised by Bienenstock [1995], who proposed the concept of synfire braids, where multiple conduction delays would correct for one another to enable synchronous representations to emerge. Around the same time, Tank and Hopfield [1987] and Hopfield [1995] proposed that a template of appropriate conduction delays could offset particular spatiotemporal patterns, thereby serving as a powerful basis for a variety of recognition tasks. Up until this time, however, the precise mechanism by which the necessary delay-lines would actually emerge had received little attention.

Gerstner et al. [1996] subsequently demonstrated that in a model of owl sound localization, an STDP-like learning rule could naturally select the appropriate conduction delays that ensured the coincident arrival of spikes with high precision. Thus, what was initially a distribution of heterogeneous delay-lines was selectively pruned to ensure only the most relevant conduction delays persisted. Izhikevich [2006] extended the concept of synfire

braids [Bienenstock, 1995] and the delay-line templates of Hopfield [1995] to suggest that asynchronous, albeit time-locked spike patterns might represent a general format for representations in the brain. In particular, such activity was termed polychrony, and it was demonstrated that in spiking neural networks with heterogeneous conduction delays and STDP, the necessary delay-lines to enable polychronous patterns to propagate widely and reliably in the network naturally emerged [Izhikevich, 2006]. Subsequent work has examined the ability of polychronous patterns to be learned via STDP and thereby enable the recognition of two different stimuli, although this has not been explored in more complex settings such as translation invariance or under varying degrees of noise, instead focusing on highly stereotyped input patterns [Guise et al., 2014]. Finally, in Eguchi et al. [2018], spiking neural networks with STDP and heterogeneous conduction delays were investigated in the setting of images of geometric objects. Once again, however, this evaluation was performed outside the context of translation invariance.

In addition to the above work where learning took place through STDP, the emergence of selective delay-lines has also been considered through the direct modification of the conduction delays themselves. Such an approach would be consistent with a role of myelin plasticity [de Faria et al., 2021] for object representations [Bowers and Davis, 2017]. One example of relevant work is the self-organising form of Davis' word-recognition model, where weights signified the conduction delay associated with a connection [Davis, 1999]. More recent work modelled the plastic modification of delay-lines in a robotic system performing path planning in a complex environment [Hwu et al., 2018], demonstrating the potential utility of such a learning approach. In Paugam-Moisy et al. [2008], a hybrid learning mechanism was used in a machine learning task on handwritten character recognition. In their model, a 'reservoir' (i.e. recurrent population) of spiking neurons was trained via STDP, after which delay-lines projecting from the reservoir were selectively modified to generate output neurons capable of detecting particular spatiotemporal (i.e. polychronous) patterns. Once again however, their task did not consider the problem of translation invariance. While the mathematical learning rules governing myelin plasticity are not currently known [Pajevic et al., 2014, de Faria et al., 2021], STDP learning is well-established, and so in our work, we will focus on the sufficiency of the latter for forming the necessary circuits for our computation of interest.

At a more general level, a variety of spiking neural network models have been considered in the context of object recognition, albeit where heterogeneous delay-lines are generally not considered. Note that we generally take an "object" in spiking neural networks to be any stereotyped spatiotemporal spike pattern, which may be created by the modeller, or may reflect an actual image input. As an example of work in this area, Masquelier and Thorpe [2007] studied the ability of STDP to support the emergence of visual features of moderate complexity, given a network's exposure to images such as faces. These features could subsequently support downstream object recognition, and this therefore represented an image-computable model. In subsequent work, the general ability to detect a particular spatiotemporal pattern in the context of noise was demonstrated through STDP [Masquelier et al., 2008], and this principle was later extended to the detection of multiple spatiotemporal patterns embedded in noise [Masquelier et al., 2009]. Later research demonstrated that these same principles could be extended (following some relaxations of biological plausibility) to replicate deep convolutional architectures [Kheradpisheh et al., 2018]. While Kheradpisheh et al. [2018] included max-pooling operations in a bid for translation invariance, the manner by which such invariant feature responses could emerge in a self-organizing network was not explored. Evans and Stringer [2012] specifically set out to determine how pooling-like invariant feature representations could emerge in spiking neural networks through STDP. In particular, they explored the role of both continuous transform [Stringer et al., 2006] and trace learning in enabling translation-invariant features. While an important contribution to the literature, we note that the objects in this work were composed of orthogonal feature-sets, and thus activated distinct populations of neurons. Furthermore, this model did not explore extrapolating learned representations to novel locations (i.e. the form of online translation invariance we consider).

Our modelling work will therefore build on, as well as differ, from prior research in several key ways. Like the modelling work in [Gerstner et al., 1996, Izhikevich, 2006, Guise et al., 2014, Eguchi et al., 2018], we will examine the sufficiency of STDP alone to select appropriate delay-lines in our network. This will be explored in the specific setting of spatial-configuration functions that may provide an avenue toward translation-invariant object recognition (as in Davis [2010b]), but where multiple sources of noise, as well as the realities of biologically plausible spiking neural networks, will need to be overcome

for the network to robustly learn and thereby encode objects.

To summarise the contributions of this chapter:

- We carry out the first investigation of whether a biologically plausible spiking neural network can use delay-lines to achieve simultaneous online translation invariance and specificity to objects composed of a common set of low-level features. This network makes use of input features that encode spatial information in their relative spike times, but needs to appropriately strengthen and weaken synapses to solve the motivating task, as well as overcome sources of noise.
- We specifically investigate the role of STDP, a well characterised, biologically known learning rule, in solving this problem. This is in contrast (and likely complementary as we will later discuss) to the proposed role of myelin plasticity in Bowers [2017] and Bowers and Davis [2017].
- We perform an initial evaluation under the setting of within-distribution translation invariance, i.e. across locations that have been experienced during learning, confirming a core capability in the network, and helping to constrain the second stage of simulations.
- We subsequently explore how well the simulated networks solve the more challenging task of online translation invariance, evaluating recognition at spatial positions that have not been experienced during learning. We confirm that such invariance can be achieved in a biologically plausible setting, and we specifically examine the role of STDP in enabling the necessary network changes to take place.
- These simulations further enable us to determine how robust the representations in simulated networks are to:
  - The two broad categories of noise likely to affect any such approach, namely diffusion noise of the position functions, and jitter.
  - Relaxations in the parameter assumptions that support the computation of interest, namely the integration window of neurons, and the drift coefficients of the position functions.

- Together, these findings demonstrate that the proposed computation is relatively robust to sources of noise and the choice of integration window, but dependent on a sufficiently large drift coefficient. We also observe multiple lines of evidence that suggest that, under the learning setting we consider, invariance through spatial-configuration functions is likely to be a more specialized computation employed by the brain, if indeed it is found in biology at all. In our discussion, we address the potential significance of myelin plasticity for some of these conclusions.
- As a secondary outcome, we compare an emergent property of the network to experimentally measured data, namely the conduction delays of non-pruned projections. Following learning in the network, we find that the distribution of these delays approaches the estimated values in Egger et al. [2020].

4.3 Methods

The majority of techniques and model descriptions relevant to our results were introduced in the Methods section of Chapter 3, as concepts such as the differential equations of spiking neurons, and the network architectures we evaluate, are relevant to both the analysis conducted in the proceeding chapter, and the simulations we consider here. Below, we include additional methods specific to our network simulations, namely the approach to training and evaluation, hyperparameter tuning, and the performance measures we use to evaluate the object representations of a network.

4.3.1 Training and Evaluation

During training for the setting of within-distribution translation invariance, the stimulus (T or $\perp$) occurred randomly at any one of three possible locations (of a total of four possible non-overlapping object locations). These same locations were then systematically sampled after learning, during which no further STDP occurred. Training consisted of 1000 presentations of each stimulus, while evaluation data was generated from 30 presentations of each stimulus (i.e. 10 presentations at each location, 60 presentations in total). For an overview of how spikes are actually sampled for objects during training and

evaluation, please refer again to Section 3.5.3 in Chapter 3.

During training for the setting of online translation invariance, the stimulus (T or $\perp$) always occurred at the same extreme position of the receptive field (i.e. $x^h = 0$ and $x^v = 0.25$ for the T stimulus, or $x^h = 0.25$ and $x^v = 0$ for the $\perp$ stimulus). During evaluation, the object was systematically presented at this familiar location, as well as three progressively displaced locations, thereby sampling all four, non-overlapping object positions that are possible (where we note again that an object takes up 25% of the overlapping receptive fields). Thus the most distal position would correspond to $x^h = 0.75$ and $x^v = 1.0$ for the T stimulus, and $x^h = 1.0$ and $x^v = 0.75$ for the $\perp$ stimulus. Training (i.e. with STDP active) consisted of 1000 presentations of each stimulus, while evaluation data was generated from 120 presentations of each stimulus (i.e. 30 presentations at each location, 240 presentations in total).

4.3.2 Hyperparameters

The hyperparameters for the network simulations are provided in Chapter 3, Section 3.5.2.4. Here, we detail the tuning of hyperparameters that was done using simulations generated from separate random seeds to those used in the final results, and which was restricted to the setting of within-distribution (i.e. not online) translation invariance. General hyperparameters that were tuned are:

- α_- **STDP constant** : As in previous modelling work such as Song et al. [2000] and Masquelier et al. [2009], we modified the learning bias, enhancing long-term depression over potentiation via the α_- constant in order to promote overall weakening of synapses in the presence of uncorrelated pre and postsynaptic spikes. As noted in Song et al. [2000], such learning in a network promotes sensitivity to presynaptic spike timing.

- **Lateral inhibition connection probability** : This was tuned to ensure that the average firing rate of excitatory output neurons was in keeping with the lifetime sparsity required for a neuron to encode the presence of a stimulus in its binary activity, indicating sufficient competition amongst neurons. In particular, the network will need to learn to distinguish two different objects, and an average firing rate of

one spike per alternative presentation ensures that the neural activity is appropriate for a binary code.

- **Weight of background excitatory synapses** : This was tuned to ensure a noisy, albeit sub-threshold, average membrane voltage in the excitatory output neurons. Such a sub-threshold mean membrane voltage is both a characteristic property of networks that can perform coincidence detection [Kempter et al., 1998, Rossant et al., 2011, Kanev et al., 2016] and is consistent with observed biological states [Anderson et al., 2000].

- **Maximum strength of excitatory feed-forward synapses** : This was increased to ensure that two approximately coincident spikes were sufficient to cause a post-synaptic spike, consistent with the settings that support coincidence detection as explored in Rossant et al. [2011], and to ensure tolerance to some degree of temporal non-coincidence. In our simulations, we explore the significance of changing this value, and thereby the integration window of neurons.

4.3.3 Performance Measures

4.3.3.1 Information Theory

We make use of information theory techniques from [Rolls and Milward, 2000], which were originally developed to analyze the information content found in the firing rates of neurons. These information theory techniques were in turn derived from Shannon information theory [Shannon, 1948].

In brief, these methods measure how much information a particular neural response carries about a stimulus by determining how specific and consistent the response is to the stimulus. In the work of Rolls and Milward [2000], a neuron can carry full information about a stimulus by firing with a high rate every time the stimulus is presented, and with a low rate or not at all when other stimuli are presented. As we are interested in neurons that spike in response to a coincident input, we instead look at the information carried in the binary activity of a neuron. Specifically, the response of a given cell can be either on or off, depending on whether it spikes at least once during the stimulus presentation

period. If a neuron is consistently active for every presentation of a particular stimulus, and not during the presentation of another stimulus, then it carries maximal information.

More formally, the response r of a given neuron is either 0 or 1 (depending on whether it spiked at least once within the interval of a stimulus' presentation), and a stimulus o is either T or $\perp$. By examining the response of a neuron over trials where the same stimulus is presented, we can determine $P(R = r|O = o)$, the conditional probability of a particular response given a particular stimulus. The information that a given neuron's responses provide about a particular stimulus is then formalized in the following equation:

$$I(o) = \sum_r P(R = r|O = o) log_2 \frac{P(R = r|O = o)}{P(R = r)} \tag{4.1}$$

where $P(R = r)$ is the marginal probability distribution of a response, i.e. the probability of a response without concern for the particular stimulus presented. The maximal information a neural response can carry about a stimulus is $log_2(m)$, where m is the number of stimuli. In our study $m = 2$, and therefore the maximal information that a single output neuron can have is $log_2(2) = 1$ bit. Finally, as in Rolls and Milward [2000] and Elliffe et al. [2002], we take as the final information value for any neuron its maximum value for any one stimulus, rather than the average across the stimuli (the mutual information).

4.3.3.2 Linear Classifier Accuracy

To measure the amount of information carried in the activity of multiple neurons, we fit linear classifiers (logistic regression) on the binary activity of neurons to predict the stimulus that was presented on a separate evaluation data-set. The parameters of the linear model are determined using Scikit Learn's [Pedregosa et al., 2011] limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm with an L_2 penalty on the size of weights.

4.4 Results : Within-Distribution Translation Invariance

We begin by presenting results where object recognition requires within-distribution translation invariance. In this task-setting, the network is trained on three locations for each object, and evaluated on those same three locations. Our simulations begin in this setting as it is important to demonstrate that the network can actually solve this simpler task, as it represents an additional manner in which we would expect translation invariance to emerge in neural representations, and the setting under which previous work in biologically plausible translation invariance is often primarily evaluated (e.g. Elliffe et al. [2002], Stringer et al. [2006], Evans and Stringer [2012]).

In addition to the above motivation, evaluating on the task of within-distribution translation invariance will enable us to establish a parameter space within which to evaluate out-of-distribution (i.e. online) translation invariance. In particular, the network will be provided with strictly more information in the setting of within-distribution translation invariance, and as such, any parameter space that we identify as being unable to support recognition in this setting should not be capable of supporting the harder task of online invariance. The boundaries we identify from within-distribution invariance are particularly useful because they capture computational limits in a concise manner, in the sense that they are not dependent on the choice of the object location, as all locations considered are within the training distribution. In other words, the task performance can be described by a single metric for a given parameter setting. In the online-translation invariance setting however, there is always the possibility of translating more (or less) in order to change the difficulty of the task, making it more challenging to establish clear boundaries.

4.4.1 Performance Across Parameter Space

We now examine the performance of within-distribution translation invariance, where we exclusively consider this in the setting of a drift coefficient of $\mu = 32$ms, and an integration window of 8ms. In particular, we wish to establish a viable parameter space for our later, more difficult task, and so we begin with this more optimal setting. For our first results, we examine the information content of cells in two parameter settings, where

the jitter is $s = 1$ms, and the diffusion coefficient is either $\sigma^2 = 0$ or $\sigma^2 = 40$. Figure 4.1 provides the ranked information content of single cells, where the cells are agglomerated from the dense networks simulated across five random seeds. We note that each network contains 20 output neurons, and so this "ensemble" network has a total of 100 output neurons; the reason for considering such an ensemble network will be clarified in our next section. As can be seen, information in the network increases dramatically following learning through STDP, where the information metric is evaluated across presentations at the three locations for each object. In order to have a high level of information, a neuron must therefore reliably activate whenever its object is presented, regardless of the location, and remain silent when the other object is shown. This is made challenging by the fact that the objects are composed of the same features, simply with a different spatial arrangement. As such, all of the input-feature neurons will spike for each stimulus presentation, regardless of the object that is presented. In Section 4.5.2, we will explore in greater detail how STDP supports the task of object-specific representations.

While we see an increase in information, it is also clear that as diffusion noise increases (Figure 4.1b), even to only a moderate amount ($\sigma^2 = 40$), and even on this simpler task of within-distribution translation invariance, information levels begin to drop. As a reminder, increasing the diffusion coefficient will result in the position functions for each feature deviating from the assumption that they are linear and of the same gradient. Indeed, across the entire ensemble network, no neuron carries perfect information (1 bit) about the stimulus across all presentations. While many neurons are clearly selective for a particular stimulus, the finding that they do not carry perfect information about the object presented is consistent with many experimental measures of neural coding (e.g. [Rolls et al., 1997]). As such, a general trend will be that highly robust representations at the level of the output layer will emerge only across multiple cells.

In addition to the single-cell information ranks, Figure 4.1a shows the number of predicted single-object alignments from our analysis in Chapter 3, Section 3.6.2. We highlight again that the expected number of alignments is not an intrinsic property of the network, but is a measure dependent on the alignment-tolerance ε, which we chose on a heuristic basis in the previous chapter to be $\varepsilon = 1$ms. In particular, it is assumed that alignments far

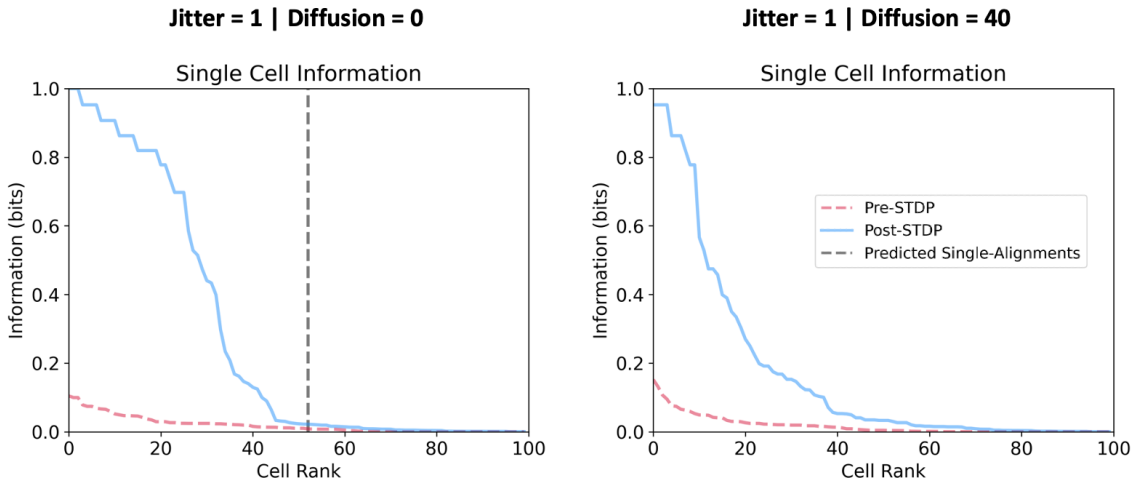


Figure 4.1: *Samples of Single-Cell Information on Within-Distribution Translation Invariance.* Provided is the ranked information in single cells across an ensemble of five dense networks, where each dense network has 20 output neurons, and is initialized and simulated with a different random seed. Results are shown for two different hyperparameter settings. The dashed grey line represents the predicted number of single-object alignments given the analysis in Section 3.6.2, and an alignment-tolerance of $\varepsilon = 1$ms. We show this in the left plot, where the network more closely aligns with the assumptions made in our analysis.

outside this tolerance are unlikely to have any significance for neurons receiving them, although which precise alignments neurons do actually make use of is dependent on their own internal properties and the consequences of learning. As can be seen in Figure 4.1a however, the heuristic we adopt appears to offer a reasonable approximation of how many neurons in the network will benefit from learning, and thereby ultimately deviate from pre-STDP information levels.

While the above results demonstrate performance at two hyperparameter settings, how does information in the network vary more generally as a function of the diffusion coefficient and jitter? In Figure 4.2 (top row), the maximum single-cell information is presented in a heat-map for a diverse range of hyperparameter values, where we use an exponential search across jitter values to determine the point at which information is completely degraded. The maximum-single cell information is determined separately for each of the dense, 20-neuron networks, where the value displayed is then the mean across 5 random seeds. As can be seen, moderate levels of information are available after learning across a variety of hyperparameters, but this degrades at high levels of diffusion noise and jitter.

In Figure 4.2 (bottom row), the accuracy of linear classifiers is shown. In the current setting of within-distribution translation invariance, the classifier is fitted to the output neuron responses that follow presentations at the three training locations. Its accuracy is then determined based on presentations at those same locations, albeit on a separate, simulated data-set of neural activity. As in the information theory heat-map, a classifier is trained for each dense network, and the result at each grid-point represents the mean across 5 random seeds. Once again, we observe that information in the network degrades as noise is introduced, although it is also clear that by combining information from 20-output cells, the classifier can still achieve high accuracy (e.g. greater than 95%) even as the maximum single-cell information in the network drops. Finally, we include the equivalent results, albeit before STDP learning has taken place; the performance in this setting demonstrates the crucial role of STDP in ensuring that the output neurons learn to respond to inputs in a stimulus-specific (yet translation-invariant) manner.

Combining information across multiple neurons is a natural way to reduce sensitivity to noise, and is an important approach in a setting such as spiking neural networks (e.g. Gerstner et al. [1996]). Even in the total absence of jitter in the input features, our output neurons are subject to significant amounts of noise, such as stochastic background inputs, and the random intrusion of spikes from other feature neurons that are co-active. In other words, even in the setting of 0 jitter and 0 diffusion noise, the networks we consider are not noiseless systems. Figure 4.3 shows the accuracy of a linear classifier as the number of output neurons it receives as features is gradually increased, where the output neurons are taken from an ensemble of 5 dense networks. It is clear that in a lower noise setting, only a handful of output neurons need to be used as features to achieve ceiling accuracy. As the amount of noise in the system increases however, information from more output neurons must be combined to achieve robust performance. We also note that even if information is combined across the 100 output neurons of a network ensemble, and even in a low noise setting, the accuracy of a classifier does not meaningfully differ from chance before STDP has shaped the output representations.

For many of our results, we will consider information derived from a linear classifier as reflecting multi-cell information. However, we note that it is naturally possible that further

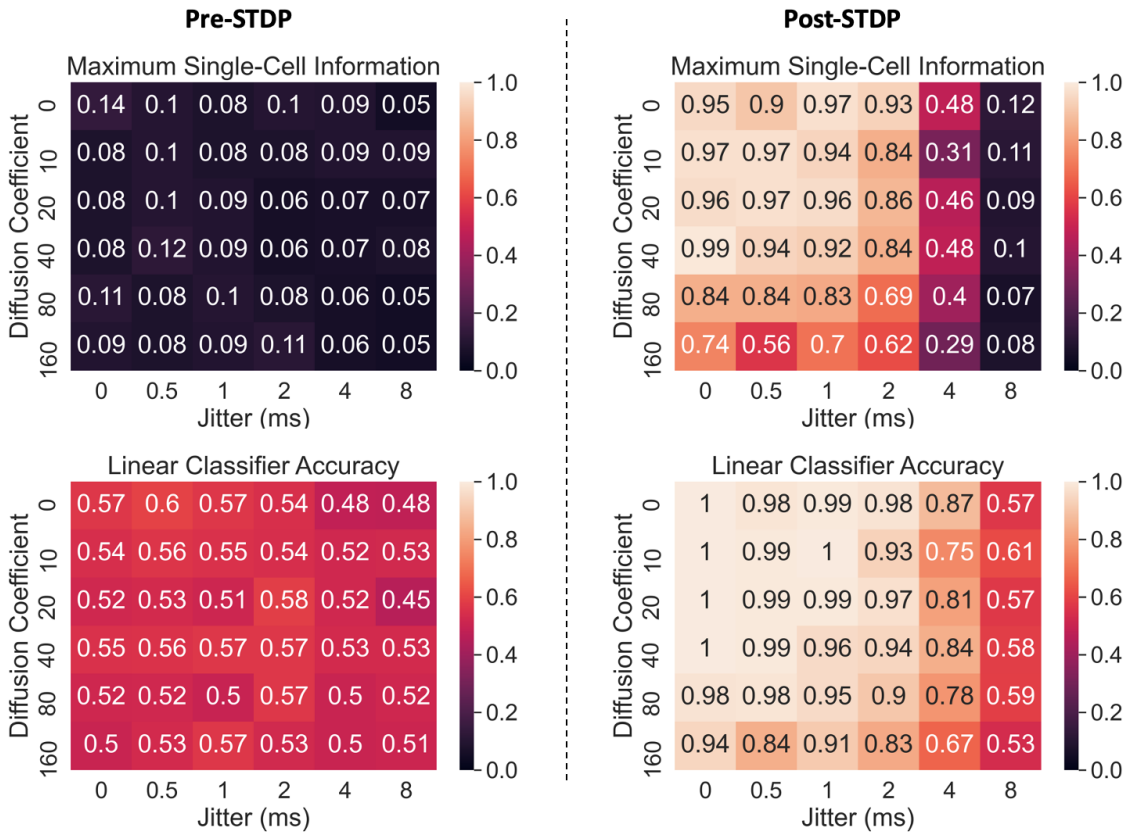


Figure 4.2: *Maximum Single-Cell Information and Classifier Performance Aross Diverse Hyperparameter Settings.* The top row shows results from the maximum single-cell information for a network; in the bottom row, the accuracy of a classifier fit to the output activity of a dense network is presented. Both result types are presented as the mean across 5 random seeds. The two columns correspond to representations in the output layer either before or after STDP.

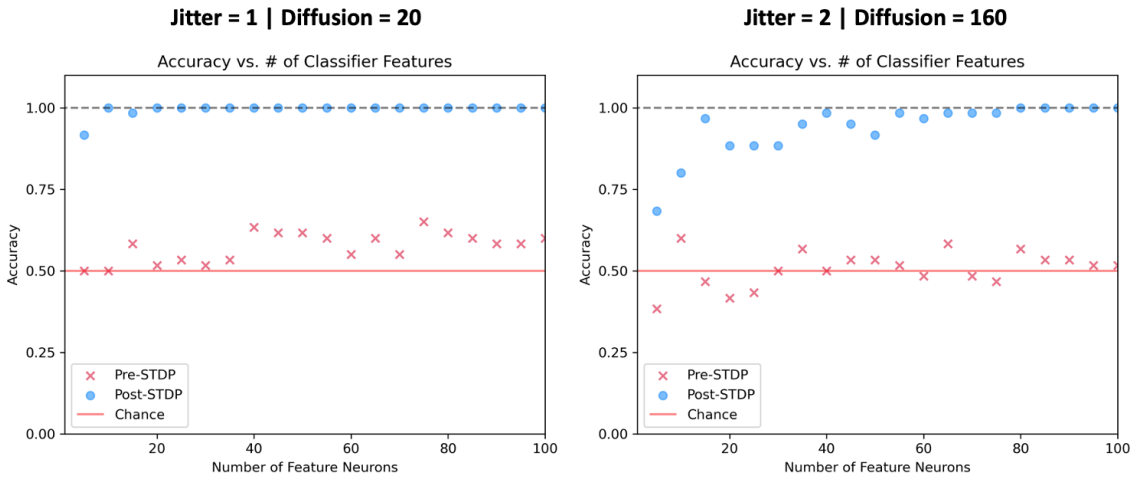


Figure 4.3: *Accuracy of Linear Classifier as a Function of the Number of Output Neurons Used as Features.* Shown is classifier performance as a function of feeding in a subset of all 100 output neurons of a network ensemble (i.e. activity combined across the dense networks generated from five random seeds). We provide results for two different hyperparameter settings.

downstream neurons which are integrating this information might be highly selective in their response profiles. For simplicity, we examine multi-cell information in our output layer with logistic regression fit with BFGS, rather than by training e.g. a perceptron using biologically plausible learning rules.

4.4.2 Dense Networks as Approximations of Larger, Sparse Networks

The above classifier results suggest that robustness can emerge on the task of within-distribution translation invariance by combining information across multiple output neurons. It is also clear, however, that to be biologically plausible, the number of such output features cannot increase in an unbounded manner. What then is a reasonable upper limit? Furthermore, we have so far shown some results where we averaged performance across dense, 20-output neuron networks, and other results where performance was considered for a single agglomerated "ensemble" network. Finally, we argued in Chapter 3 that dense networks could serve as a reasonable approximation of larger, sparse neural networks, but would enable us to efficiently simulate results across a diverse range of hyperparameters, as well as simplify the analysis of network properties. We now demonstrate that the performance of dense networks across random seeds can indeed serve as a useful approximation of larger, sparse networks.

In Figure 4.4 (top row), the single-cell information of the dense networks, combined across 5 random seeds as an ensemble network, is once again provided. Beside this, the accuracy of a classifier trained on the ensemble network (i.e. all 100 neurons) is provided, where this classifier setting is equivalent to the final end-point in Figure 4.3. As can be seen, there is a dramatic increase in the robustness of the classifiers given a full ensemble network for fitting (cf to Figure 4.2).

In Figure 4.4 (bottom row), we show the performance of larger spiking neural networks, where connections are sparse. In particular, the sparse architecture is the same as the smaller dense neural networks with the exceptions that i) each input feature is associated with a population of 100 neurons, rather than 10, for a total of 200 input feature neurons, and therefore 100 rather than 10 input assemblies, ii) feed-forward synapses form with a probability of 0.1, rather than 1.0, iii) lateral synapses form with a probability of 0.01, rather than 0.5, and iv) the output population consists of 1000 neurons, or 500 sub-sampled neurons, where relevant.

For this larger, sparse network, performance is first shown for single-cell information given a subset of 500 output neurons, which corresponds to a setting with a similar number of predicted single-object alignments as in the dense ensemble setting. As can be seen in Figure 4.4 (left top vs left bottom), the dense ensemble network offers a reasonable approximation of the sparse network. We include in this figure the predicted number of single-object alignments in the sparse network, using the separately derived equation from Chapter 3. Once again, this offers a reasonable reflection of the proportion of neurons that increase their information content following learning.

In Figure 4.4 (right, bottom), we provide the accuracy of linear classifiers trained on a population of 1000 output neurons from a sparse neural network. As before, the dense ensemble network and sparse network appear approximately equivalent.

Why is the ability for the dense networks to, once taken as an ensemble, approximate the behaviour of a larger, sparse network useful? We demonstrate this result because it will enable us to consider both the performance of a network when constrained to a relatively limited, more efficient neural substrate, as well as performance when the population of neurons dedicated to the computation in question expands significantly. In particular,

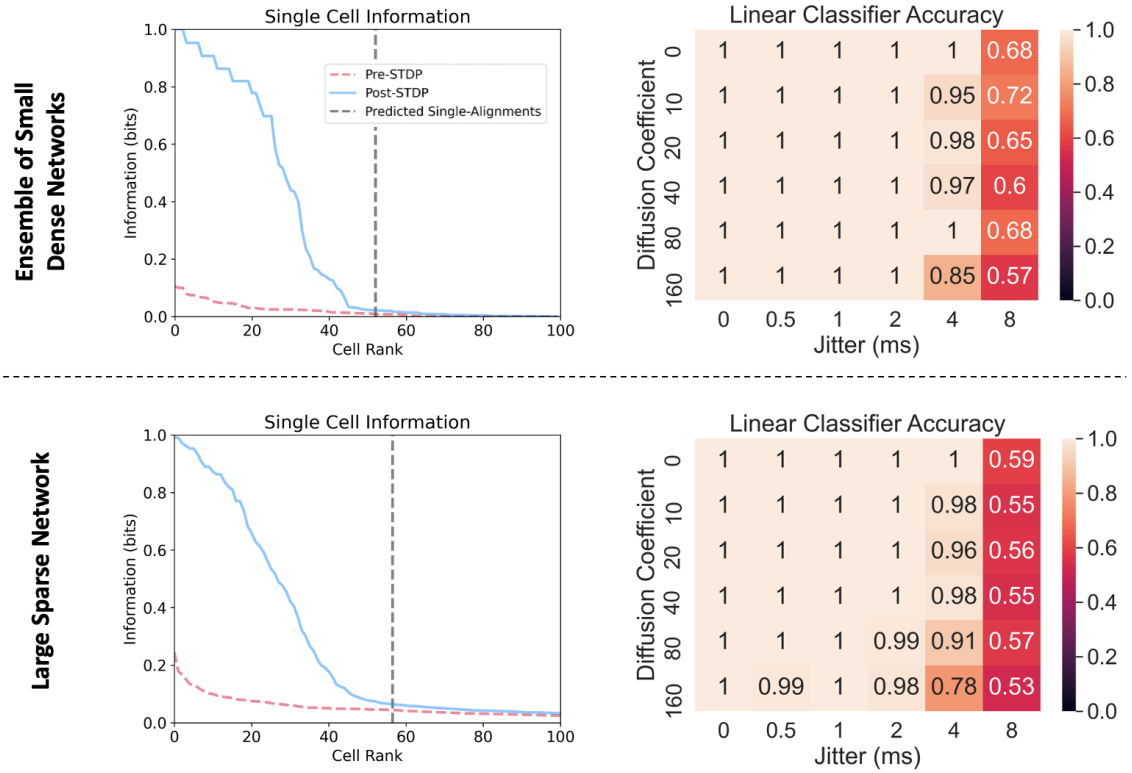


Figure 4.4: *Approximate Equivalence of Sparse Networks and an Ensemble of Dense Networks.* In both the setting of single-cell information, and the accuracy of linear classifiers fit to a population of responses, we observe that an ensemble of smaller dense networks can offer a reasonable approximation of the performance seen in larger, sparse networks. For the single-cell information in the sparse network, we limit the x-axis to the top-100 most informative neurons. In both single-cell information settings, the results correspond to the parameter setting of $s = 1$ and $\sigma^2 = 0$.

a single, 20-output neuron dense network is roughly equivalent to a 100-output neuron sparse network, which might correspond to the input neurons projecting predominantly to a single cortical mini-column (as discussed in Chapter, Section 3.6.2). It may be, however, that the input neurons project more widely, for which a reasonable upper limit might be 5-10 times the population of input neurons, corresponding to the performance we observe for the ensemble network. By being able to analyse our results under both these settings, we can efficiently estimate the performance of the system as it scales, including determining the limits of robustness given fewer neurons. Finally, we note that performance is actually slightly higher in the ensemble network, particularly for classification accuracy. This is likely a result of correlated features in the sparse network, which are less useful for the linear classifier. Such correlation of output-neuron responses would itself be due to the shared population dynamics resulting from, for example, lateral connectivity. As such, this serves to reinforce the notion that the performance of the ensemble network represents an upper limit of robustness, assuming a sufficiently large population of neurons is dedicated to the recognition task we consider.

Given the above, we can return to the classifier results from Figure 4.4 to estimate the hyperparameter space that we will evaluate for our more challenging, online-translation invariance task. The results suggest that robust outputs can emerge even under the largest levels of diffusion noise that we consider, although we note that diffusion noise is considerably less problematic for the within-distribution setting. In particular, even if the object functions are highly variable with respect to the absolute position of an object, output neurons could simply learn to respond to the particular spatiotemporal pattern that occurs at each location, even if there is no consistency between these. In the online-translation invariance setting, robustness will require that the spatiotemporal pattern of an object is consistent across multiple locations, yet it is precisely this consistency that diffusion noise will degrade.

The results also suggest that a jitter of 4ms or greater is likely to represent a significant challenge, particularly for higher levels of diffusion noise. This is an interesting finding, as many experimental results have demonstrated both *in vitro* [Berry et al., 1997], and *in vivo* [Reinagel and Reid, 2000, Kara et al., 2000] that neurons are capable of precision

on the level of 1-2 ms of jitter, and this is consistent with the level of jitter considered in previous modelling studies [Masquelier et al., 2008, 2009]. As such, 1 and 2ms of jitter will represent our primary evaluation settings, under which we will examine performance across the full spectrum of diffusion noise (0 to 160). For the strongest performing networks in online-translation invariance, we will also assess performance at 3 and 4ms of jitter. Together, these results will help establish the ability of the proposed computations to tolerate noise from both jitter and diffusion.

4.5 Results : Online Translation Invariance

In the above section, we confirmed that our spiking neural network can solve the relatively straightforward task of object-specific representations that are translation invariant, where testing occurs only at locations that have been observed during training. We also established the approximate equivalence between an ensemble of dense neural networks and larger, sparse neural networks, where dense networks will form the basis of all of the following simulations. Finally, the above results helped establish a narrow parameter space that we can evaluate in detail, which we will use to determine where the proposed computations begin to fail on the primary task.

We therefore turn to the much more challenging, and more interesting, task of online translation invariance. That is, having undergone learning at restricted locations, such invariance requires that the same object can be recognised at other, novel locations where the object has not been seen before. In the simulations presented below, we train the networks with the object shown at only a single location. We highlight again that we evaluate this under a setting where the objects are composed of a common set of features, albeit with different spatial arrangements.

In our particular simulations, each object (the T or $\perp$, although we note again that features are represented by the spike times of neurons, and not actual images) is presented at the bottom of the receptive field at training time. At evaluation, it is then either presented at the same location (i.e. within-distribution recognition), or one of three other, non-overlapping locations. These locations correspond to translating the object by a multiple

of the size of the object (25% of the receptive field). The object can therefore be translated by a total of three lengths relative to its own size, before it begins to fall outside of the receptive field that we consider.

4.5.1 Observing Online Translation Invariance in the Base Setting

We are interested in establishing the settings under which online-translation invariance begins to fail in our proposed system, i.e. the boundaries between solving this task and failure. In addition, we would like to determine the role that STDP plays in actually making this task solvable. As such, we begin our primary evaluation in the setting that our analysis in Chapter 3 determined is both biologically plausible, as well as computationally optimal for the given task. In particular, this corresponds to the setting of a drift coefficient of $\mu = 32$ms, and an integration window of 8ms. Later results will demonstrate that our analysis in Chapter 3 was correct in predicting this as the optimal setting, however for now we turn to assessing the core question of if, and how, this system can solve online translation invariance.

In Figure 4.5, we present the single-cell information of output neurons, where this information is evaluated from 30 presentations at each object location, including the single location at which training has occurred (for a total of 120 presentations of each object). To carry perfect information, an output neuron needs to respond whenever a particular object is presented at both the familiar training position, as well as the three novel locations, whilst ensuring it does not respond to the other object in any instance. These results demonstrate that in the relatively low diffusion setting, there is not a dramatic change in the amount of information from our within-distribution setting (cf Figure 4.1). This is consistent with the notion (which we will later explore in greater detail), that with low diffusion noise, the object function will be relatively consistent across absolute positions, and so online invariance is largely equivalent to the task of within-distribution invariance.

Figure 4.5 also demonstrates, however, that in the setting of $s = 2$ and $\sigma^2 = 160$, single-cell information is significantly degraded, and few neurons achieve more than 0.5 bits of information in their individual responses. As previously noted, single-cell information will therefore be unreliable in many of the settings we consider, yet a neural system can

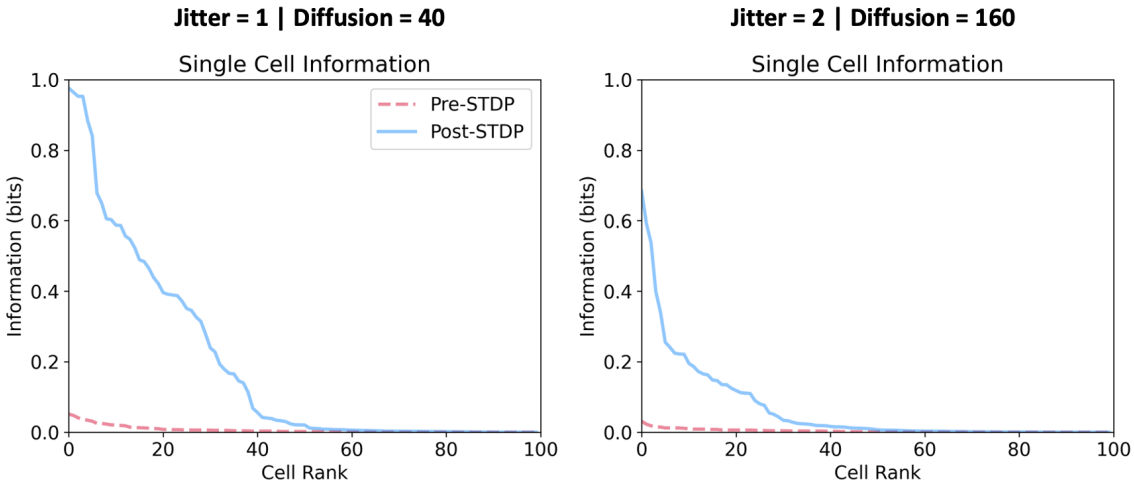


Figure 4.5: *Samples of Single-Cell Information for Online Translation Invariance.* Provided is the ranked information in single cells across an ensemble of five dense networks. Results are shown for two different hyperparameter settings.

make use of the activity of a population of neurons (which we consider through the use of a simple linear classifier) to determine the object that is currently presented. To enable an intuition for the magnitudes of diffusion we consider, Figure 4.6 provides demonstrative examples of typical spatial-configuration functions obtained with a diffusion noise of $\sigma^2 = 40$ and $\sigma^2 = 160$.

In addition to the observation that single cells may carry limited information, the information theory metric we have so far used does not tell us how performance varies as a function of the translation magnitude. In particular, we would expect that in the setting of non-zero diffusion noise, larger translations will be associated with greater changes in any given object function, and therefore a progressive drop in the robustness of a network's representation.

To therefore assess the ability of multiple cells to carry information about an object's identity, as well as the effect of larger translations, a linear classifier is fitted to neural activity from the first location in the receptive field, corresponding to the location where all STDP learning has taken place. On a separate data-set of simulated network activity, its performance is then evaluated at this familiar location, as well as the three progressively displaced positions.

Figure 4.7 (left column) demonstrates the performance of classifiers trained on a single

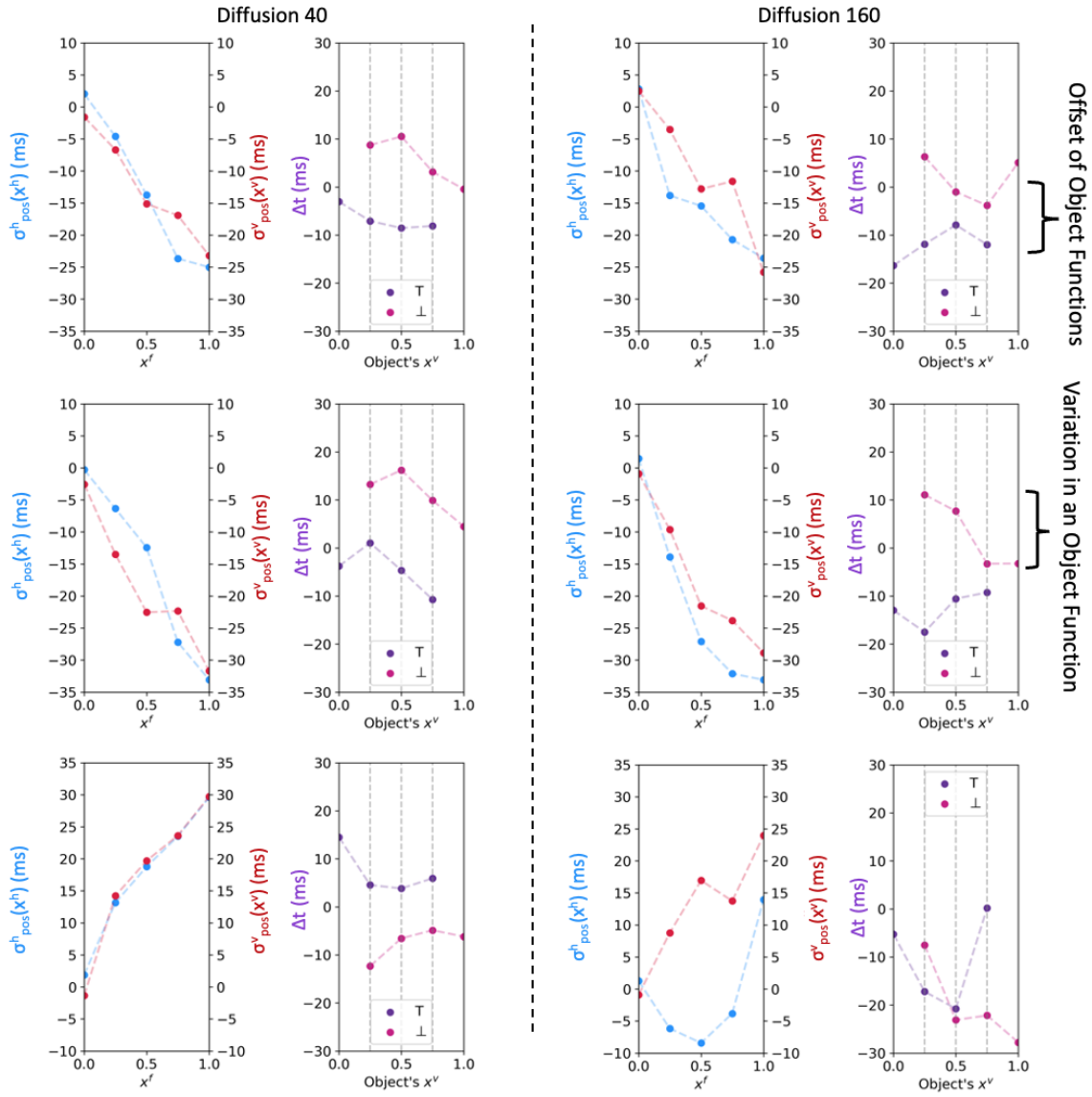


Figure 4.6: *Sample Spatial-Configuration Functions.* Provided are demonstrative examples of position functions (left in each pair) and their associated object functions (right in each pair) with $\mu = 32$ms, and $\sigma^2 = 40$ or $\sigma^2 = 160$. On the right-hand side, we include annotations to convey the concepts of the offset of object-function means, and the variability of object functions, two measures that we will use in our later analyses.

dense neural network, where the presented results are the mean across 5 random seeds. As expected, performance degrades as a function of the distance of translation, except in the setting of very low ($\sigma^2 = 0$ or $\sigma^2 = 20$ with $s = 1$ms) diffusion noise. Furthermore, a higher jitter ($s = 2$ms) results in a moderate degradation in performance. This degradation is notable, however, in that even in the setting in which the object functions deviate very little from true linearity ($\sigma^2 = 20$), accuracy begins to drop at the farthest translation point. We also highlight that these results are in the setting that our analysis in Chapter 3 determined to be the optimal one for robustness (i.e. $\mu = 32$ms, integration window of 8ms), demonstrating the significant challenge that this task presents.

In Figure 4.7 (right column), we see that this picture changes significantly when the linear classifier is fit to the ensemble network. By combining information across a larger population of neurons, close to perfect accuracy can be achieved even in the challenging setting of $s = 2$, and considerable amounts of diffusion noise ($\sigma^2 = 80$), where the latter corresponds to a significant deviation from a linearity assumption. Interestingly, however, even in the setting where ensemble networks are leveraged, a diffusion of 160 proves too significant a perturbation for the maximum translation that we consider.

These results indicate that, without leveraging significant neural representations, and thereby aggregating information across a large population of neurons dedicated to the computation of interest, robust recognition is generally not achievable at large online translations. We previously identified in Chapter 3 that the relatively low probability of delay-line alignments was evidence that the computation we are exploring is likely to represent an uncommon basis of neural representations. Our finding that significant neural populations must be leveraged for robustness provides the first result from our simulations that further aligns with this view. In other words, while the proposed computation is attractive for the robustness to both online invariance and form specificity that it can bring, it is also expensive from the point of view of the neural hardware that must be dedicated. As such, the contribution the computation makes at a behavioural level may be proportional to how familiar a stimulus is, a possibility that we will return to in our Discussion.

It is worth highlighting the relationship of our results in Figure 4.7 to experimental work in

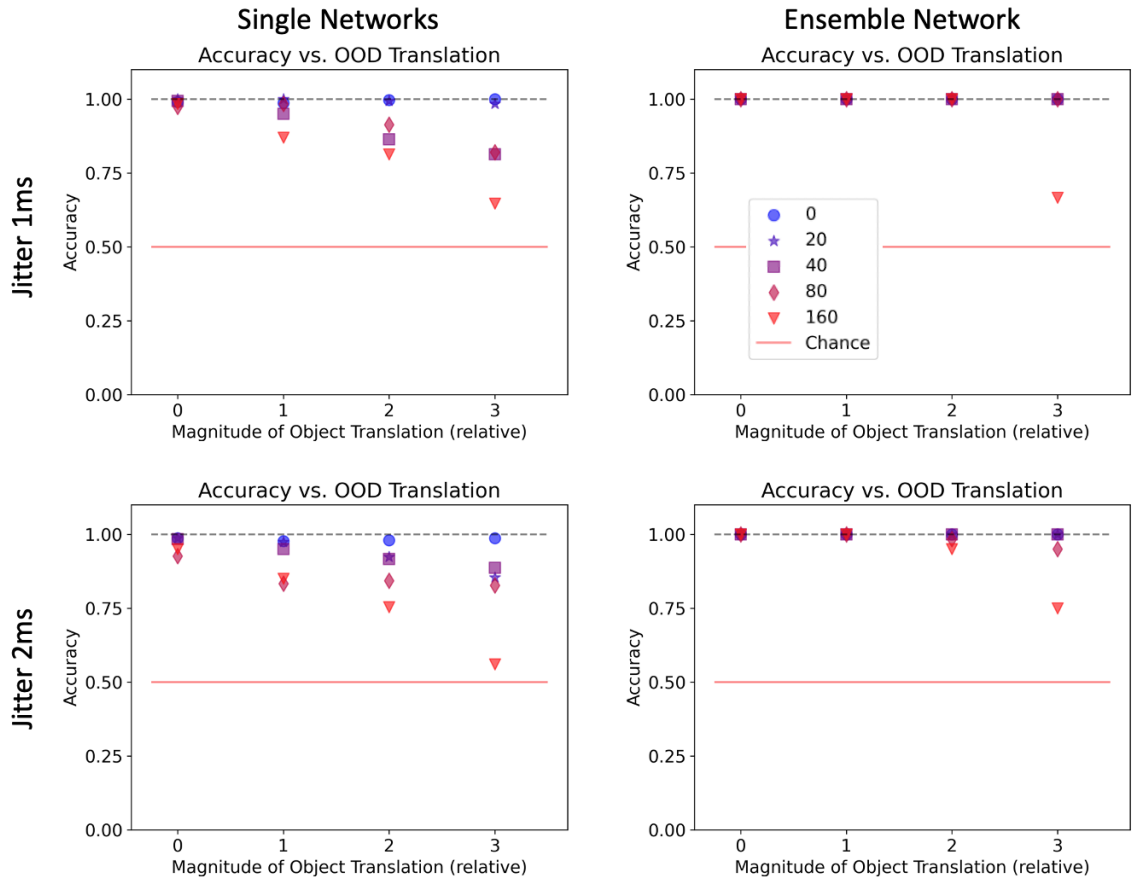


Figure 4.7: *Accuracy of Linear Classifier Representations as a Function of Online-Translation.* Provided is the accuracy of linear classifiers fitted to neural responses corresponding to 0 translation. These classifiers were then evaluated at separate neural responses corresponding to this location, as well as progressively displace positions. The markers in the legend indicate different levels of diffusion noise. As the object fills 25% of the receptive field we consider, displacements correspond to non-overlapping translations of the object relative to its own size, where a translation of $3\times$ is the maximum possible. Classifiers in the left column are fit on single dense networks of 20 output neurons, where the presented result is the mean across 5 random seeds. In the right column, classifiers are fit on ensemble networks. The two rows correspond to different magnitudes of jitter. OOD = out-of-distribution (i.e. online) translations.

humans on online translation invariance such as Blything et al. [2021]. In their work, the stimuli were 3D, geometric objects that filled 5×5 degrees of visual angle. Study participants learned an object at a particular retinal location, before the objects were presented at translated positions of 3, 6, 9, or 18 degrees, and the accuracy of object recognition evaluated at the new locations. In their primary evaluation, accuracy at 9 degrees was 94% ($\pm$8SD), where this translation roughly corresponds in our setting to a $2\times$ relative displacement. Even at 18 degrees of translation (corresponding to a larger translation of $3.6\times$ their object width, in contrast to our $3\times$ translation), humans still achieved 84-89% accuracy. Importantly, Blything et al. [2021] were careful to use objects composed of common parts in diverse spatial arrangements, not unlike the simplified objects we consider. Our results therefore represent an interesting hypothesis for how humans achieve such a high degree of online translation invariance, and whilst leveraging representations that are sensitive to spatial form.

While we do not explicitly model reaction times or the effects of prolonged exposure, it is also worth highlighting that our results demonstrate performance improvements as the linear classifier has access to the responses of more output neurons. Owing to the fact that the network activity is sparse and progresses over the presentation window, time would be required to gradually integrate additional output neuron responses. At a coarse level, our results are therefore consistent with the finding by Blything et al. [2021] that online translation invariance in humans benefits from prolonged exposure times, particularly at extreme displacements, whilst still being possible following more brief exposures.

It is also important, however, to note some key differences, and in particular, simplifications that we have made in order to evaluate performance in a biologically plausible, spiking neural network. Whereas both our task and that in Blything et al. [2021] considered distinguishing two classes ("A" or "B" in their study) corresponding to a chance performance of 50%, they consider a more realistic, challenging setting where 12 different objects exist in each of these classes. In earlier work, humans were found to perform well above chance on even harder tasks, such as circa 80% accuracy at displacements of 13 degrees, where chance was around 17% [Bowers et al., 2016]. It is for this reason (i.e. the simplicity of our stimuli) that we emphasize instances of ceiling performance in

our model, even though humans do not achieve 100% accuracy following extreme online translations.

Furthermore, our network is of course not image-computable (i.e. cannot process actual images), and uses idealized "objects" composed of two features in different configurations. In contrast, Blything et al. [2021] identified that CNNs with sufficient pre-training, or which were augmented with global average pooling, could achieve comparable performance to humans when presented with the same images. Indeed, it is clear that on the one hand, our portrayal of pooling as inducing a bag-of-features state is a simplification given properties of CNNs such as the hierarchy of processing, overlapping receptive fields and (where relevant) skip connections. On the other hand, it remains unclear the cost by which CNNs achieve their invariance in terms of robust object representations [Kamath et al., 2021], including to object form [Baker et al., 2018, Brendel and Bethge, 2019, Geirhos et al., 2019, Malhotra et al., 2020, Geirhos et al., 2021]. Whilst the authors were careful to select 3D stimuli that, at a conceptual level, are composed of common features-sets in unique configurations, it may be possible that the CNNs were still able to identify low-level features that are object-specific (e.g. due to a particular viewing angle), and thus solve the task without relying on the full configuration of the geometric parts. Furthermore, it is unclear the importance that weight sharing played in achieving comparable translation invariance to humans, and whether a more biologically plausible implementation [Pogodin et al., 2021] would enable equivalent performance.

Finally, we have noted that results from our simulations suggest that the computations we explore are potentially relatively specialized, and would operate alongside standard pooling-based methods for conjunction detection. A possible consequence of this is that, at a behavioural level, the influence of spatial-configuration functions would be most prominent for particularly familiar stimuli. Such a conclusion is at odds, however, with the notion that these computations could prove useful for the online translation invariance of novel stimuli (as used in Bowers et al. [2016] and Blything et al. [2021]). We will return to this point in our main Discussion.

In summary, our work differs from prior experiments, as well as models, in several key ways. While our results indicate that the networks we consider are broadly consistent with

human performance, and can achieve this in a biologically plausible manner, we therefore do not make claims about whether our networks represent a more faithful model of human perception than CNNs.

4.5.2 The Basis of Robust Representations

In the above section, we demonstrated that representations can emerge in biologically-constrained networks that enable online translation invariance, even in the setting of considerable levels of diffusion noise (i.e. deviations from linearity in the position functions). We further found that the performance of our network bears some comparison to the performance of humans on a similar task. In the following, we analyse the precise role that STDP plays in enabling robust representations through learning. We subsequently consider the significance of object-function properties and delay-line alignments for the amount of information that representations carry.

4.5.2.1 The Role of STDP for Robustness

We have argued that, in line with previous modelling work (e.g. Gerstner et al. [1996], Izhikevich [2006]), STDP should be able to select the appropriate delay-lines that offset the spatiotemporal spike patterns associated with object functions, resulting in a preference for synapses that are associated with the coincident arrival of signals. Moreover, we noted that in Chapter 3 that at the outset, connectivity in the network will be dense, and as such, output neurons will be subject to large amounts of noise from afferent synapses carrying spikes, even when these synapses are not useful for solving the task at hand. This noise corresponded to a small afferent interspike interval (ISI), and we highlighted that to perform temporally precise computations, the network would likely need to prune synapses. In the setting we currently consider (i.e. an integration window of 8ms), we estimated that approximately 50% of synapses would need to be pruned to bring the afferent ISI to an acceptable level (i.e. from 5ms to 10ms).

Figure 4.8a confirms that, as a result of STDP, weights are dramatically pruned, resulting in an approximate weight sparsity of 0.13. With significantly fewer synapses associated with moderate-to-strong weights, the afferent ISI increases to approximately 38ms, far

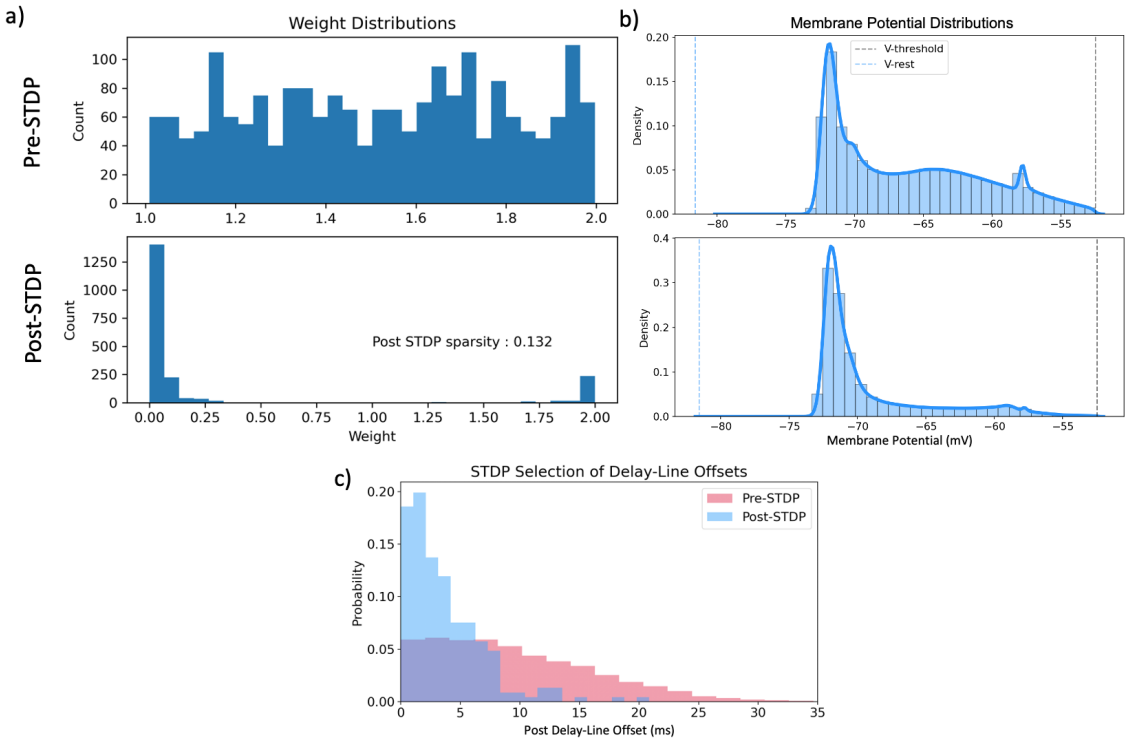


Figure 4.8: *The Role of STDP in Supporting Robust Object Representations.* a) Following learning, STDP results in a dramatic shift in the distribution of synaptic weights from one that is uniform, to one that is sparse and associated with only a small subset of strong connections. Note the difference in the x-axis origin. b) This reduction in the number of (irrelevant) afferent synapses is associated with a shift to a more tightly peaked membrane potential distribution, such that neurons are less likely to respond following the intrusion of random spikes. The presented membrane potential is combined across the 20 output neurons of a single network, across the entire simulation period. Results from (a) and (b) are taken from the setting of $s = 1$ms, $\sigma^2 = 40$. c) Rather than simply reduce noise, STDP also selectively strengthens and preserves delay-lines that correct for the spatiotemporal patterns with which objects spike. Results are taken from the setting of $s = 1$ms, and combined across the diffusion levels $\sigma^2 = 10$ through $\sigma^2 = 160$.

in excess of the integration window (8ms), and thus supportive of temporally precise computations taking place [König et al., 1996]. In keeping with this reduction in noise, the membrane potential of neurons becomes peaked in a narrower region (Figure 4.8b). As such, STDP plays a crucial role in reducing the level of noise in the network, and thereby supporting the use of computations that rely on temporally precise coincidence detection.

Is the role of STDP to simply reduce noise however? To quantify the effect on delay-line offsets, we first define a few quantities. Any given object function can be summarised

by the expected value it will take across all four presentation locations (i.e. the average spatiotemporal pattern across all positions). We call this quantity the object function mean, which we evaluate as the expectations $\mathbb{E}(\Delta t_T^a)$ and $\mathbb{E}(\Delta t_\perp^a)$, where a indexes the object functions associated with a particular input assembly. For every output neuron, we then determine if they share a pair of moderate or stronger (e.g. $w_i \geq 1.0$) weights to a particular assembly. If so, we calculate the final temporal offset after the associated delays (i.e. d_i^h and d_i^v) are applied to each object function mean, providing us with the "post-delay-line offset". We report the post-delay-line offset associated with the object for which the output neuron responds preferentially (corresponding to the offset that the output neuron seemingly makes use of), and otherwise randomly select one of these if it has no preference.

Figure 4.8c demonstrates that, through STDP, there is indeed a dramatic concentration of delay-line offsets that correct for the associated object function's spike pattern. In other words, STDP is not simply pruning synapses randomly, as this would result in no shift in the distribution. Instead, it selectively favours those delay-lines that result in the approximately coincident arrival of signals at an output neuron. As such, STDP contributes to robust representations both through a general reduction in noise, as well as its ability to specifically favour delay-lines associated with the coincident arrival of spikes.

4.5.2.2 The Significance of Object Functions and Delay-Lines

Above, we explored the role of STDP in enabling the network to develop robust object representations. We now seek to quantify the relationship between the object functions and delay-lines associated with an output neuron, and the robustness of that output neuron's representations.

We begin by examining the significance of an object function's form. To do so, we reuse the concept introduced in the previous section of an object function mean. We are then interested in the absolute difference between the means for the two object functions associated with any given spatial-configuration function (and thereby any given input assembly), which we quantify with $|\mathbb{E}(\Delta t_T^a) - \mathbb{E}(\Delta t_\perp^a)|$. The expected significance of this

offset is that, as it increases, a learned representation for one object is less likely to overlap with, and thereby respond to, the representation for the other object.

The other quantity we consider is the variation of an object function as a result of changes in the absolute location of an object, which we quantify with the minimum standard deviation of the object functions (i.e. $\min(\sqrt{\mathrm{Var}(\Delta t^a_T)}, \sqrt{\mathrm{Var}(\Delta t^a_\perp)})$). The expected significance of this quantity is that as it grows, representations learned at one location are less likely to generalize to others. The intuition for both the object function offsets and variability are given in Figure 4.6 (right-side annotations).

Figure 4.9 confirms the importance of these quantities for robust representations. In particular, for any output neuron in the network that has a pair of strong afferent synapses from an assembly, we look at the object functions associated with that assembly. The object function offsets are plotted as a function of the single-cell information in the output neuron in Figure 4.9, while the standard deviation of the least variable object function is represented in the error bar.

As expected, strong synapses are not sufficient for a robust representation. If these are associated with a minimal offset of the mean object functions, or the object functions are associated with large variability (Figure 4.9a, bottom-left corner), then the output neuron will not carry robust information about the input stimulus. In contrast, if the function offsets are larger, and the object functions show minimal variation across absolute positions, then an output neuron with strong associated weights will generally carry high amounts of information about the object stimulus. Importantly, even in high diffusion settings (e.g. $\sigma^2 = 80$ or $\sigma^2 = 160$), object functions that can satisfy these requirements exist, although they are also less common than in low diffusion settings.

In addition to the nature of the object functions, we would also like to examine the significance of the post-delay-line offsets, where this quantity was defined in the previous section when we examined the effects of STDP. We would expect that the smaller this offset is (i.e. closer to 0), the more robust the representation will be to sources of noise, as even if jitter causes spikes to arrive with a sub-optimal offset, a sufficiently large integration window (here 8ms) should still be able to integrate these inputs. If the offset is relatively large however, then the spikes are more likely to fall outside of the integra-

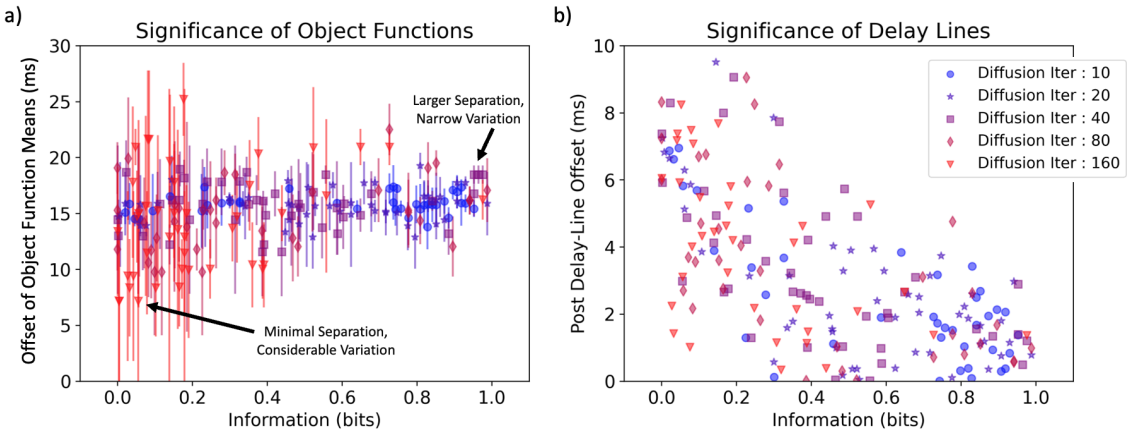


Figure 4.9: *The Relationship Between Object Function Properties, and Post-Delay-Line Offsets, With Single-Cell Information.* a) The relationship between the offset of object function means, and the information level of an output neuron with strong connections to the associated assembly. Also shown is the standard deviation of the least variable object function (error bars). b) The relationship between the post-delay-line offset of an output neuron with strong connections to a particular assembly, and the associated information level of the output neuron. Results are presented from the setting of $s = 1$ms.

tion window following jitter. Like before, we examine this quantity for output neurons that share strong weights with a particular assembly. The results shown in Figure 4.9b confirm our prediction, where we see a clear relationship between a small post-delay-line offset, and the amount of information that an output neuron can carry (corresponding to the robustness of its response). As with optimal object functions however, a good delay-line offset is not sufficient, as these may be associated with poor object functions. In addition, other factors (such as lateral competition), may prevent the output neuron from responding to every presentation of its preferred object.

4.5.3 The Failure Case : Drift Coefficient of 0

As noted in Chapter 3, if the drift coefficient is 0, the input neurons will carry no information about the spatial arrangement of the features, and it will be impossible to distinguish them when they are presented in the receptive field under consideration. This corresponds to the setting of standard pooling, demonstrating how the proposed computations can capture this often useful (if potentially limited) operation as a special case.

We note, however, that with a drift coefficient of 0, but high amounts of diffusion noise, position functions could emerge that might encode sufficient spatial information so as to

support distinguishing objects composed of a common feature set. This corresponds to dropping the assumption that the spatial-configuration functions are structured, and have some shared, monotonic bias (even if they can deviate from this linear bias significantly). While this might operate reasonably well for within-distribution translation invariance (i.e. where a separate spatiotemporal pattern can be learned for every object position), it is unclear how successful this approach would be in the setting of online-translation invariance. At the limit of increasing numbers of input assemblies, it is guaranteed that random position functions will emerge that are both monotonic and of a consistent sign. However, we have sought to establish an estimate of the number of afferent input neurons that we would expect under both biological and computational constraints. In particular, Section 3.6.2 of Chapter 3 estimated that a significant increase in the number of afferent connections that we consider would be potentially out of keeping with connection-sparsity in the brain, and would result in disruptive levels of input noise. Given this, our question is whether the ultimately limited population of input neurons we consider is sufficient to observe completely random spatial-configuration functions that support solving the task of online-translation invariance.

Figure 4.10a demonstrates that, as expected, the setting of 0 drift and 0 diffusion noise carries no information about an object, and performance is at chance both at the location at which all learning has taken place, and to novel locations. Interestingly, and as predicted, high levels of diffusion noise can actually enable object recognition at training locations (i.e. within-distribution). Importantly however, with the quantity of input assemblies we consider, these do *not* result in well-aligned position functions that can support online translation invariance. In Figure 4.10b, we confirm that the source of information given high diffusion noise is a result of increasing separation of the object functions, which are most effective when they are also associated with minimal variability of the object function (i.e. as in the setting of $\mu = 32$ms). Note that we truncate the x-axis at 0.3 to make the trend visible, and thus the amount of single-cell information across all four positions is minimal.

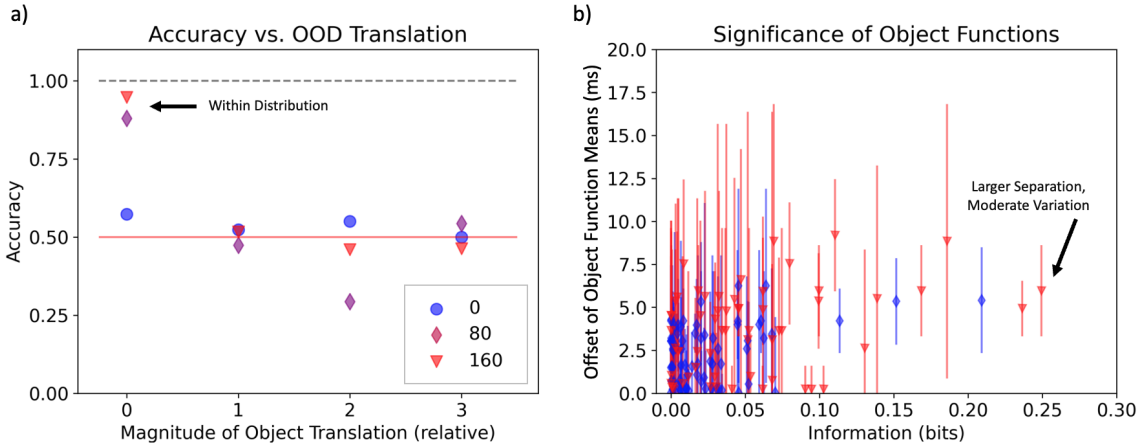


Figure 4.10: *Translation Invariance in the Setting of No Drift ($\mu = 0ms$).* a) Classifier accuracy in the setting of 0 drift, but at three levels of diffusion noise (indicated by the legend), and as a function of presentation location. Results are from classifiers fit to single (rather than ensemble) networks, but we found that the latter did not measurably change the observed trend. b) The relationship between the offset of object function means, and the information level of an output neuron with strong connections to the associated assembly. Also shown is the standard deviation of the least variable object function (error bars). Note the truncated x-axis.

4.5.4 The Significance of the Integration Window

Our analysis in Section 3.6.3 of Chapter 3 suggested that with a drift of $\mu = 32$ms, as well as the other assumptions of our network (such as the number of afferent synapses at the outset), the optimal integration window would be approximately 8ms. As the integration window decreases, output neurons would be less able to integrate learned spatiotemporal offsets in the context of noise, either due to jitter, or due to changes in an object function as a result of diffusion noise. Conversely, as the integration window increases, output neurons are more likely to respond to spikes associated with multiple, otherwise temporally unrelated assemblies. Furthermore, a larger integration window can erode object specificity, as delay-lines may result in the arrival of signals such that the integration window receives both spikes in its prescribed range, regardless of the spatial arrangement of features.

To evaluate this analysis with our simulations, we trained and assessed separate spiking neural networks, where the maximum weight of the network ($w_{\max}$) was varied so as to result in an estimated integration window of 4, 8, 12, or 16ms, as described in the previ-

ous chapter. When doing so, we corrected the uniform range of initial weights to ensure that the mean initial weight did not change from 1.5. Consistent with our analysis, Figure 4.11 demonstrates that in both low (left) and high (right) diffusion settings, the integration window of 8ms appears to strike the optimal balance of integration and coincidence detection. As discussed in Chapter 3, such a value also aligns well with several experimental estimates of the integration window of neurons, although we once again note that there is significant diversity in how to define the integration window.

More importantly, even in our own setting, the definition we use of the integration window is a simplification. For example, we enabled our analysis by defining the integration window as a discrete, all-or-nothing effect over a given temporal window. In reality, temporal offsets of different magnitudes can still have different influences on the probability with which an output neuron spikes, even if they both fall within the range of the integration window. Furthermore, our idealized simulation in Chapter 3 would suggest that the integration window as we have defined it is unchanging, regardless of a neuron's state. In actuality, this is only true so long as the membrane potential associated with the arrival time of spikes is fixed; as this will naturally vary in the context of a network subject to noise, so too will the integration window. As such, even a very large estimated integration window (e.g. 16ms) can be associated with stimulus-specific information. In spite of this more complex reality however, the results in Figure 4.11 demonstrate that the simplifying assumptions we made were sufficiently reasonable so as to support accurate predictions of our results.

Our analysis in Chapter 3 was motivated by an understanding of information at the level of individual cells. While the above results demonstrate that the integration window of 8ms is indeed optimal in such a setting, does this hold when we consider information based on the response of multiple neurons? More importantly, how robust is the network representation when we consider classifiers trained on multiple neurons, and therefore how dependent are the proposed computations on a "Goldilocks" integration window?

To assess this, we once again simulated separate networks associated with different levels of $w_{\max}$ so as to vary the estimated integration window, before assessing the performance of classifiers trained on the resulting representations. In Figure 4.12, we see that with a

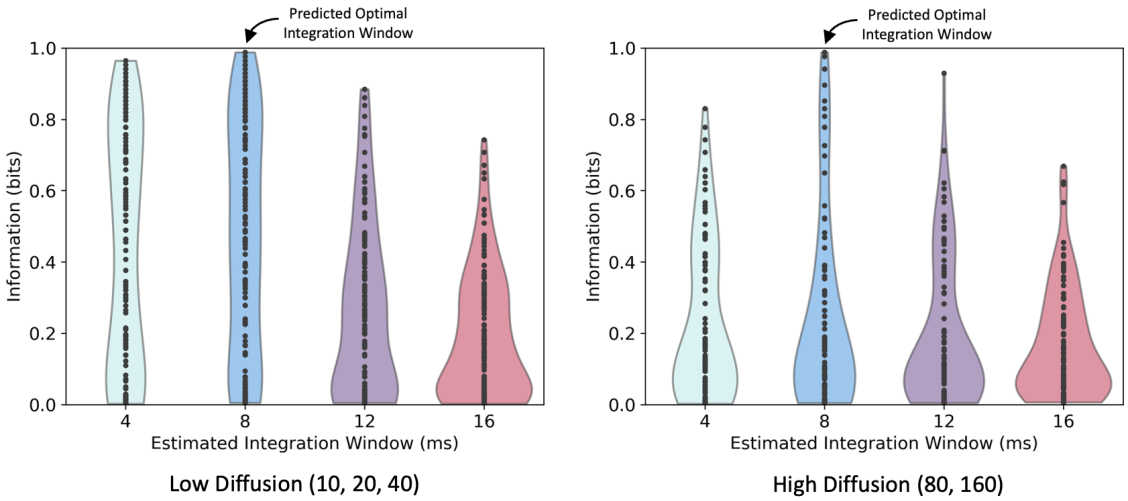


Figure 4.11: *The Effect of the Integration Window on Single Cell Information.* In the setting of $\mu = 32$ms and $s = 1$ms, we evaluate the effect of the integration window on the amount of single-cell information across all possible object positions. Shown are violin plots (smoothed distributions), along with the actual data-points (centre of each axis). Results are combined across seeds and a given set of diffusion values, which are either low (left) or high (right). When combining results across seeds and diffusion values, the top 0.5 proportion of the most informative cells for each setting are included so as to avoid the visualized distributions becoming overwhelmed by low-information cells. As well as the general optimality of the integration window of 8ms, the results also demonstrate that as one moves from the low to high diffusion setting, the comparative advantage of an especially narrow vs. broad integration window shifts.

very large estimated integration window (16ms), performance is indeed slightly worse in the setting of classifiers fit to single dense networks, and in the context of minimal diffusion noise. When the classifier is fit to the population of an ensemble network, however, these deficiencies are resolved, and more importantly, we actually observe an *increase* in accuracy in the challenging setting of $\sigma^2 = 160$ and the maximal translation magnitude (cf Figure 4.7).

How could the larger integration window improve performance in this setting, when Figure 4.11 clearly demonstrates that at the single-cell level, 8ms is optimal even in the high diffusion setting? While some of the single cells in the latter case carry high levels of information across all locations, the classifier does not identify a priori which neurons are robust to online translations, as it is fitted given only the activity of the network at the first location. As such, robustness to online translations does not factor into its weighting of individual output neurons (i.e. features). When translations do occur, the narrower integration window of 8ms will result in many neurons being unable to accommodate the change in their associated object functions, obscuring the signal from the small subset of neurons that carry high information across all positions. In the setting where the integration window is much larger, individual neurons carry less information, but for those that do carry some diagnostic value (and for which the classifier has therefore weighted as useful features), these response profiles will be less sensitive to changes in the object function when translations take place. Consistent with this view, we also observe that given a narrower integration window, classifier performance is particularly sensitive to online translations (Figure 4.12, bottom row).

While not a possibility that we explore, various signals might be available to help the linear classifier identify output neurons that will be robust across large translations, without needing to actually observe their activity (and therefore the object) at these novel locations. For example, the robustness of an output neuron's activity to small translations in a local area (i.e. spatial jitter) might carry a weak signal about how robust its response is to larger translations. Such an approach might thereby enable the features in setting such as an integration window of 8ms to be used more effectively.

Taken together, we find that both at the single-cell and ensemble-level of information,

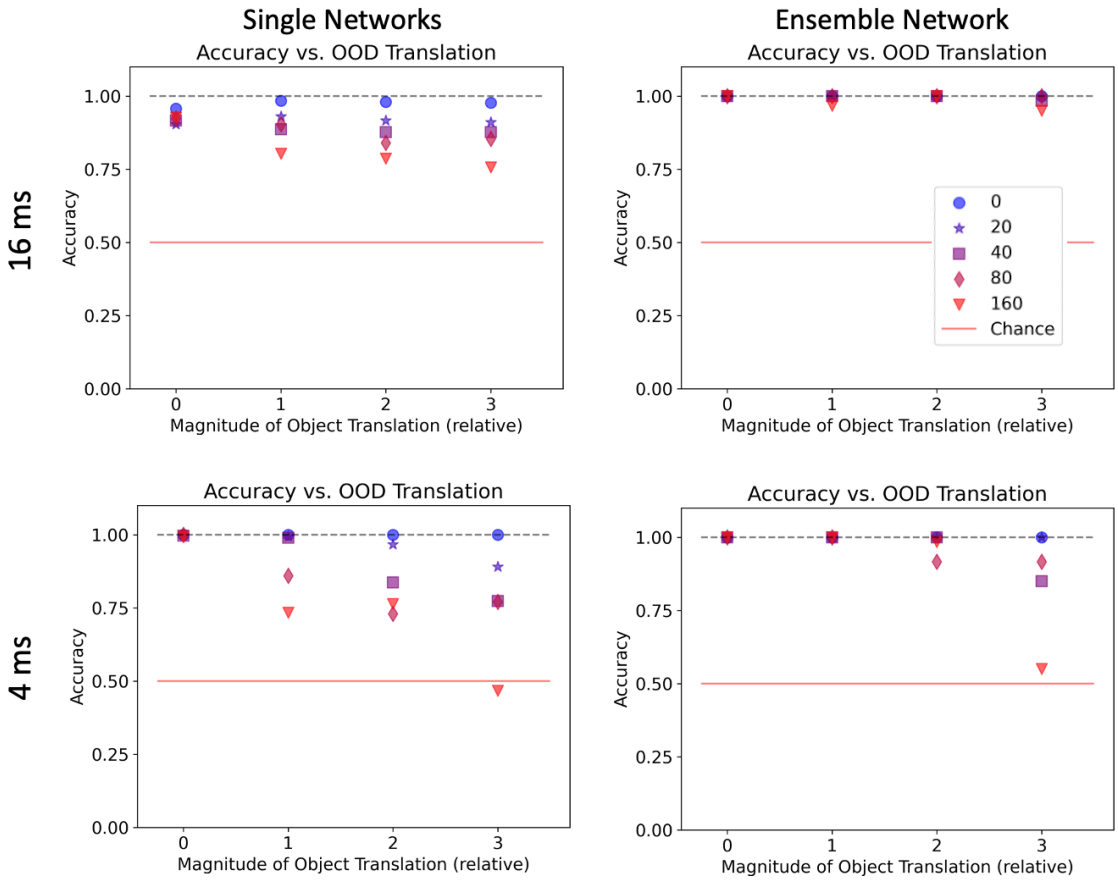


Figure 4.12: *Classifier Accuracy Under Different Integration Windows.* Presented are the accuracies of linear classifiers where these are fit to single dense networks (left; mean accuracy across 5 seeds), or an ensemble network (right). Results are shown in the context of an estimated integration window of 16ms (top row) or 4ms (bottom row). In all instances, $s = 1$ms; NB that accuracy degraded for the setting of $\sigma^2 = 160$, $s = 2$ms, even with an integration window of 16ms and an ensemble-based classifier. The markers in the legend indicate different levels of diffusion noise.

the computation we propose is not beholden to a precise integration window to enable encoding robust information about the input object. Instead, depending on the nature of information encoding in the system, as well as the amount of diffusion noise, different integration windows can offer particular computational strengths. We note that the experimentally derived integration windows discussed in Chapter 3 differ partly due to the fact that they represent estimates under different physiological conditions, such as the level of background neural activity, or the presence of feed-forward inhibition. This presents with it the possibility that the brain might adjust the integration window to optimally respond given a particular input setting. Such a tailored approach might be informed by properties such as the activation sparsity of output neurons, thereby determining whether the integration window should be narrowed (neurons are responding too often), or widened (neurons are responding too rarely).

4.5.5 Performance Given Higher Amounts of Jitter

One of the motivations of our work is to establish the parameter settings within which the online translation invariance can no longer be solved by our proposed networks. We have already established that a diffusion level of $\sigma^2 = 160$, when associated with the maximal translation that we consider, is generally incompatible with robustness, unless the output neurons have an especially large integration window. Section 4.4 estimated that jitter above 2ms was likely to represent a challenging setting for robust object representations. To investigate this in our current setting, we evaluate the optimal network ($\mu = 32$ms, integration window of 8ms) in the setting of 3 or 4ms of jitter. As expected, we observe in Figure 4.13 that as jitter increases, the accuracy of the network drops. In the setting of 4ms, such a performance decrease is observed in spite of using the ensemble network, and even in the context of low levels of diffusion noise (e.g. diffusion=20). As such, these results are consistent with our earlier estimate, which is that if these computations are to be found in the brain, then neurons will need to operate in a regime where jitter is constrained to approximately less than 3ms. As noted earlier however, these levels of noise are both consistent with some experimental measures of jitter [Berry et al., 1997, Reinagel and Reid, 2000, Kara et al., 2000], as well as the assumptions and sensitivity analyses of previous modelling studies [Masquelier et al., 2008, 2009]. We also include the object

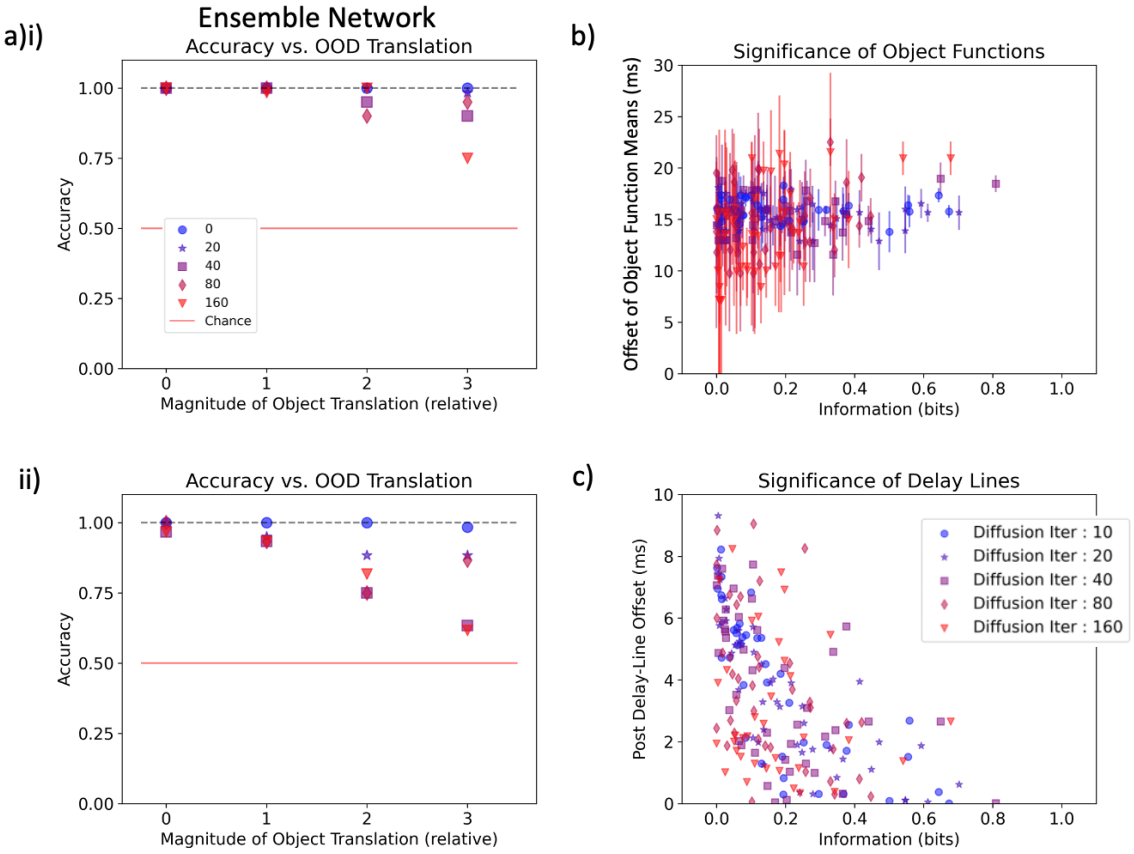


Figure 4.13: *Classifier Accuracy Given Higher Levels of Jitter.* Presented are the accuracies of linear classifiers where these are fit to ensemble networks in the setting of $s = 3$ms (a.i) or $s = 4$ms (a.ii). The markers in the legend indicate different levels of diffusion noise. Also presented are results from the analysis of the object functions (b) and post-delay-line offsets (c) given $s = 3$ms. These results demonstrate both the decrease in single-cell information values, as well as the particular importance of optimal object functions and precise-delay-line alignments when jitter increases.

function and post-delay-line offset analyses (Figure 4.13b and c), demonstrating that with more jitter, optimal values become even more important for single-cell information.

4.5.6 A More Conservative Drift Coefficient

In Chapter 3, we estimated that the drift coefficient in biologically observed position functions could be around 32ms, and that this would be an approximately optimal value given the constraints we considered. We also noted, however, that we would evaluate a smaller drift coefficient ($\mu = 16$ms) in order to assess the effect of a more conservative estimate, as well as to determine the sensitivity of the proposed computations to the magnitude of this parameter. Performing such an evaluation also serves as an additional assessment

of our analysis of the integration window; with a 50% drop in the drift coefficient, we would expect the optimal integration window to show a similar contraction (i.e. to 4ms), as larger integration windows would result in difficulty distinguishing objects that differ only in their feature configurations.

We therefore perform simulations with networks in which $\mu = 16$ms, observing that the smaller drift coefficient does not support particularly robust representations (Figure 4.14). This is especially pronounced when we consider the accuracy of classifiers to out-of-distribution translations. More encouragingly, we find that the prediction that the integration window of 4ms would be optimal at the single-cell information level holds true. Likewise, an integration window of 16ms results in the obliteration of most information in the network.

As expected from our analysis in the previous chapter, a smaller drift coefficient prevents the network from benefiting from larger integration windows, and therefore constrains the overall robustness of the system to noise from both jitter and diffusion. It is worth highlighting the importance of this observation. In the setting of $\mu = 32$ms, relatively robust representations could be found either at the single or multi-cell level regardless of the integration window value. In contrast, we *do* observe sensitivity of the proposed computations to the drift coefficient. This implies that, unless biological systems can implement drift coefficients on the scale of 32ms or greater, then the ability of the computation to support robust, online-translation invariance will be limited. At the same time, we observed in Chapter 3 that drift coefficients which are much larger than this could prove problematic, as these are associated with a reduction in the probability that delay-line alignments take place. As such, our results suggest that there may be something of a "Goldilocks" value where the drift coefficient is concerned. This represents the second result from our simulations that online-translation invariance through spatial-configuration functions is likely an uncommon operation performed in the brain, at least if implemented through the learning mechanisms that we explore.

As a final note, while the setting of $\mu = 16$ does not present a particularly robust solution to our opening task, and $\mu = 0$ cannot resolve it at all, these settings could still prove useful for encoding information. In particular, while networks in these settings cannot reli-

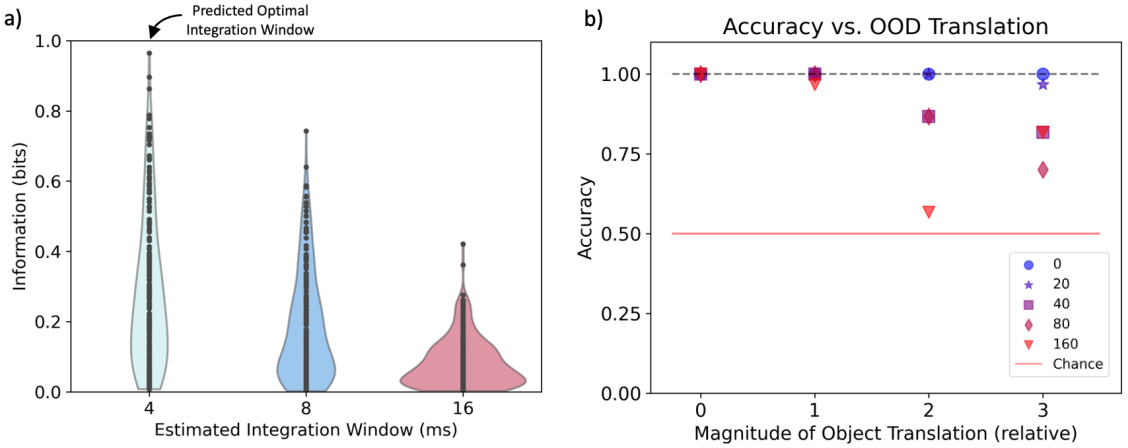


Figure 4.14: *Performance Given a Smaller Drift Coefficient ($\mu = 16ms$).* a) In the setting of $\mu = 16$ms and $s = 1$ms, we evaluate the effect of the integration window on the amount of single-cell information across all possible object positions, and with results aggregated across the diffusion coefficients 10 through 160. b) Accuracy of an ensemble-based classifier, given an integration window of 4ms, and $s = 1$ms. We also assessed this performance with the other integration windows, but accuracy was no better. The markers in the legend indicate different levels of diffusion noise.

ably distinguish two objects composed of a common set of features, output neurons *would* be able to recognise if an object with these features was present or not - in other words, a bag-of-features detector within the receptive field in question. As already discussed at length, this is the computation that max-pooling representations support, and when leveraged at multiple levels of hierarchy and in overlapping receptive fields of varying size, this appears sufficient for many challenging visual tasks. As such, while our results suggest that we require $\mu \approx 32$ms to robustly solve the particular task with the networks we consider, this need not imply that spatial-configuration functions with considerably lower μ will not be useful. Moreover, the setting of $\mu = 16$ may represent an interesting compromise, where output neurons respond in an approximately bag-of-features manner, but for which the probability of their spiking to a particular object is modulated to some degree by the spatial configuration of features. How multiple drift coefficients could interact in a single network, potentially combining the strengths of pooling and spatially-sensitive conjunction detection, represents an interesting question for future research.

4.5.7 Preserved Delays and the Relation to Experimental Work

For our final simulation result, we wish to examine an emergent property of our simulated circuits, and in particular how this compares to the known properties of biological networks. While STDP has been leveraged so as to enable output neurons that robustly encode object identity, do the subsequent models resemble biological networks according to another measure? To answer this, we examine the distribution of delay-lines following learning. In particular, we used the upper limit of the delay-lines observed in Egger et al. [2020] to establish a uniform distribution of delays as an agnostic (albeit clearly simplified) initial delay-line distribution, as might be observed in infancy. We note in particular that the rats that were the basis of their data corresponded in age to young adults, and thereby the age after which the majority of synaptic pruning has occurred in rodents [Semple et al., 2013]. The question of interest is then whether the change in this distribution is consistent with the delay-line distribution observed in their work, where they estimated this as a log-normal distribution with mean 3.4ms and standard deviation 2.3ms. In Figure 4.15a, we show that following learning, the delay-lines associated with non-pruned synapses do indeed appear to approach the distribution that Egger et al. [2020] measured. In particular, larger delay-lines will often be needed in order to offset the spatiotemporal patterns associated with different objects in our networks. At the same time, however, given two delay-lines that both offset a delay (e.g. 1 and 7 ms, or 4 and 10ms), the first of these will activate an output neuron first. Owing to competition in the form of lateral inhibition, such shorter delay-line pairs are likely to be preferentially strengthened, and the skewed distribution we observe is a natural consequence.

Despite some broad similarities, the distribution of delay-lines we measure is clearly an imperfect match, and in particular, we observe a comparably much higher probability of longer delay-lines. We note that the distribution from Egger et al. [2020] is based on somatosensory cortex neurons in rodents. As such, a straightforward explanation might be that the distribution measured in primate visual cortex could differ, although we are not aware of any evidence to support this speculation. On the other hand, the distribution we observe would more closely match that found by Egger et al. if the circuits we consider are embedded in networks of more standard (non-delay-line dependent) computations, where

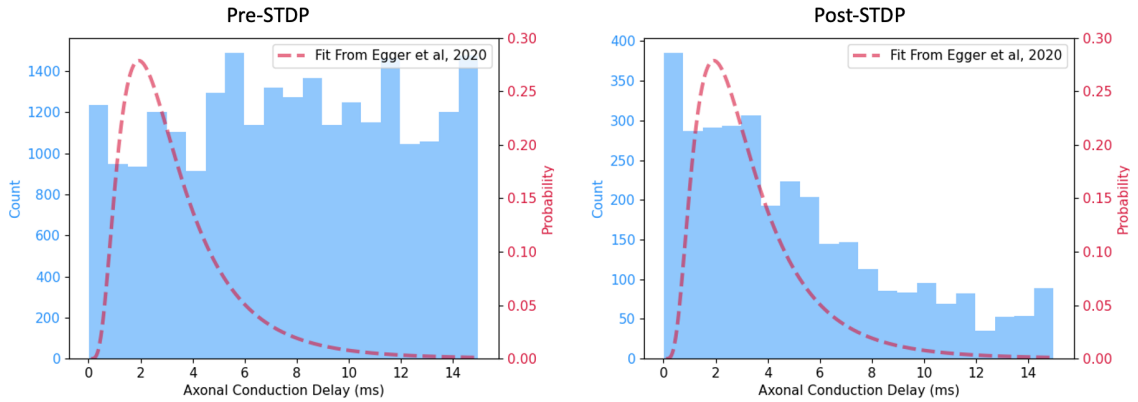


Figure 4.15: *The Preserved Conduction Delays Following Learning.* Shown is the distribution of conduction delays associated with moderately strong ($w_i \geq 1.0$) synapses before (left) and after (right) learning. Super-imposed is the estimated log-normal probability distribution from Egger et al. [2020] (mean 3.4ms, standard deviation of 2.3ms). The displayed results are aggregated from networks with $\mu = 32$ms, jitter of 1 or 2ms, and diffusion from 0 to 160. Note that the initial distribution (pre-STDP) was uniform, so any deviation from this is simply a result of noisy sampling.

shorter delays will have an outsize role. In other words, the circuits we explore would represent only a small component of the broader cortical connectivity. This represents the third, and final result from our simulations that implies that, should spatial-configuration functions play a computational role through STDP learning in the brain, they are likely to represent a specialized computation.

4.6 Discussion

In the above work, we extended the theoretical ideas and analysis from Chapter 3 to simulations of spiking neural networks. Through these simulations, we evaluated the ability of spatial-configuration functions to serve as a basis for robust object recognition in biologically plausible networks. In particular, we investigated the ability of such representations to support conjunction detection that is both translation invariant, and sensitive to the spatial configuration of features. Rather than simply identify a single point at which robustness can be achieved, we assessed performance across a range of hyperparameters, identifying that robust representations are relatively insensitive to the integration window of neurons, assuming a sufficiently large drift coefficient. As the drift coefficient parameter decreases, however, robustness on the task of online translation invariance falls

considerably. Furthermore, in the setting of $\mu = 0$, representations displaying online translation invariance do not emerge even in the presence of high diffusion noise, suggesting that spatial-configuration functions must have a shared, non-zero bias ($\mu > 0$) if they are to serve a computational purpose in biologically-constrained networks.

In addition to demonstrating the successful learning of robust representations, and identifying the parameters under which such computations begin to fail, our simulations confirmed several predictions from the analysis in Chapter 3. These included the relationship between the expected number of single-cell alignments and the number of cells that improve their stimulus selectivity following learning, as well as the optimal integration window given a particular drift coefficient. Furthermore, we observed that after learning, the conduction delays in our networks shifted from a uniform distribution to one that more closely matches those measured experimentally [Egger et al., 2020].

4.6.1 Spatially-Sensitive Conjunction Detection as a Specialized Computation

In spite of the demonstration that the computation of interest could take place in spiking neural networks subject to noise and a variety of biological constraints, both the proceeding chapter and our simulations identified a number of lines of evidence that the computation is likely to be relatively specialized, rather than the central method of conjunction-detection found within visual cortex.

In Chapter 3, this conclusion was suggested by the observation that the existence of multiple position functions with comparable monotonic form is likely to fall as the dimension of the feature-set increases. As such, it would likely be the case that the proposed computations take place alongside more typical, pooling-supported conjunction detection. Such additional feature information could result from both neurons with Poisson-like firing rate responses, as well as settings where the drift coefficient is low, following which the positional information in spatial-configuration functions would be eroded. Further evidence of the specialised nature of the explored conjunction detection mechanism emerged in Chapter 3 when we analysed the probability of delay-line alignments taking place. In particular, we considered whether random delay-lines, together with subsequent selection

via STDP, would be sufficient to observe the necessary circuits for our computation of interest. While our analysis and simulation results confirmed that this is indeed possible in certain parameter regions, the analysis also demonstrated that with higher-dimensional feature sets, these alignments can be associated with low probabilities. As such, many output neurons will be incapable of displaying sensitivity to spatial configurations, and as such, we once again concluded that only a subset of features for any particular object are likely to be encoded with spatial-configuration functions.

In our simulations, we subsequently observed that for representations to be robust to distant online translations, information would likely need to be agglomerated across a large population of neurons, indicating that a significant volume of cortex would need to be dedicated to support the most robust limits of the computation we explore. Furthermore, our sensitivity analysis identified that given a more conservative drift coefficient of $\mu = 16$ms, robustness to large translations was significantly degraded from the setting of $\mu = 32$ms. At the same time, our analysis in Chapter 3 demonstrated that the probability of alignments falls with increasing separation of the spike times associated with an object, presenting a possible soft constraint on particularly large drift coefficients. Given these constraints from both above and below, this suggests the drift coefficient may need to occupy a fairly narrow region for spatial-configuration functions to provide their full benefit to robustness.

Finally, we observed that learning in our network pushed a uniform distribution of conduction delays to more closely align with experimentally observed distributions in young-adult animals [Egger et al., 2020]. However, our distribution still demonstrated a disproportionate share of axons with moderate to long conduction times (e.g. 5+ ms). We therefore proposed that the circuits we consider might be embedded within more numerous circuits performing standard computations that are less reliant on longer delay-lines.

In summary, we identified a variety of pieces of evidence suggesting that, although possible within a biologically constrained system, the computation we have considered is unlikely to represent the dominant mechanism of conjunction detection in the primate brain. As previously noted, the networks we explore do not result in the wastage of neurons. Rather, when the necessary circuits fail to emerge owing to consequences such as

the absence of delay-line alignments, or low drift coefficients, the low-level features revert to performing a similar operation to max-pooling. In particular, a neuron will respond so long as its feature is somewhere in its receptive field, but without conveying any additional spatial information. Conjunction detection can then operate normally on these invariant representations. Similarly, we have discussed several ways in which firing-rate neurons with smaller weights could serve a complementary role, where these would contribute less spatially sensitive information about the features that are present. As already noted, such a mixed approach would be compatible with apparently conflicting accounts for firing rate and temporal codes in the brain, and the log-normal distribution of weights in cortex [Song et al., 2005, Buzsáki and Mizuseki, 2014] (as rarer, large weights are associated with more coincidence-detection properties [Rossant et al., 2011]). Furthermore, this finding might align well with the observation that current CNN models can explain a significant portion, albeit not all, of the variance of single-cell firing in primate cortex [Yamins et al., 2014, Bashivan et al., 2019].

The above findings suggest that, if implemented by the learning mechanisms we have considered, conjunction-detection that is sensitive to the spatial configuration of features is likely to have a greater impact for familiar, rather than novel, stimuli. In such instances, more learning will have taken place, likely increasing the number of relevant low-level features that are encoded through spatial-configuration functions, as well as the probability of delay-line alignments emerging. Furthermore, the brain could dedicate more neural hardware to ensure that the encoding of familiar (and therefore potentially behaviourally relevant) stimuli is robust. In our simulations, this dedication of a larger neural population was reflected in the enhanced robustness of classifiers that used ensemble networks for their input features.

In line with this possibility, there is evidence that at a perceptual level, human sensitivity to the spatial configuration of features also benefits from familiarity. In other words, representations may shift from less-spatially sensitive, pooling-based conjunction detection, to ones based on spatial-configuration functions as stimuli become more familiar.

For example, in a visual search task, humans are much faster at distinguishing familiar 5 and 2 stimuli, compared to when these are rotated 90°[Wang et al., 1994]. This is in-

teresting, as the target and distractor stimuli share low-level features in both conditions. Despite this, the ability to distinguish them based on the spatial configuration of these features only appears to be supported rapidly and in parallel across the visual field when the stimuli are well known. Similar effects have been observed for example in identifying letter T's and L's. When these letters are upright (and therefore represent a familiar configuration) they can be recognised in the periphery even when attention is engaged elsewhere [Fei-Fei et al., 2005], and identified in a search task relatively efficiently [Wolfe and DiMase, 2003]. If the letters are rotated, they are no longer as familiar, and pre-attentive object recognition in humans is closer to chance performance [Fei-Fei et al., 2005]. Interestingly, although models based on CNNs can enable impressive, qualitative matches to a variety of search asymmetry effects seen in humans, these do not currently explain the apparent efficiency with which humans can distinguish T's and L's, instead performing comparatively slowly [Gupta et al., 2021].

More generally, there is a variety of evidence that humans show sensitivity to the spatial arrangement of features for frequently encountered stimuli such as faces [Tanaka and Farah, 1993], and words [Davis and Bowers, 2004], and that these effects can extend to various types of objects following sufficient exposure [Diamond and Carey, 1986, Gauthier and Tarr, 1997]. At the neural level, Desimone et al. [1984] probed cells in the visual cortex of macaques that are responsive to stimuli such as hands and faces. Such neurons demonstrate remarkable robustness to the degradation of low-level features, or rotations that preserve the global relations of features. However, when local features are preserved while their spatial relations are disrupted (i.e. scrambled objects), the response outputs of the neurons drop significantly. We once again contrast this with evidence that CNNs can show excessive sensitivity to low-level features such as textures, and comparative insensitivity to global object shape [Baker et al., 2018, Brendel and Bethge, 2019, Geirhos et al., 2019, Malhotra et al., 2020, Geirhos et al., 2021].

With stimuli such as words, there is also evidence that the spatial representation of features is consistent with the monotonic encoding we have explored here. For example, when perceiving briefly presented pairs of words, human participants can experience letter migration errors, such that the words PEACE and PLATE might be associated with

the perception of "PLACE" (E-L or C-T migration). Such migrations can occur for letters at other positions as well, such as the presentation of BENCH and BATCH inducing the perception of "BEACH" (N-A at an adjacent position). Interestingly, such migrations can be observed as far away as two letters, but the probability of a migration falls monotonically with its distance [Davis and Bowers, 2004]. Such an effect is difficult to account for if the spatial encoding of features is inflexibly specific to each possible position in a word (i.e. spatial slots), or insensitive to spatial location (i.e. pooled representations). In other words, this implies a richer form of spatial encoding such as category slots, as proposed in prior models of word recognition [Davis, 2010b] and object recognition [Sabour et al., 2017], and explored through our simulations in spiking neural networks.

4.6.2 The Significance of Myelin Plasticity

In spite of the above considerations, it is important to note that our opening interest was not just in the basis of spatially-sensitive conjunction detection, but also the relation of such a computation to online translation invariance.

Potentially contradicting the above arguments regarding familiarity, one study found that presenting objects at a fixed retinal location for tens of thousands of trials can actually be associated with minimal online translation invariance in Rhesus monkeys [Cox and DiCarlo, 2008]. Note however that this experiment involved brief presentations that were difficult for the animals to classify even at the trained location. Limited online translation invariance in such a setting might result from the very subtle differences in the low-level features for the stimuli used. Such constraints could have limited the ability of the animals to perform well, and would also limit the relevance of the computations we consider.

More consistent with our above arguments, Blything et al. [2021] observed that for the novel stimuli they used, there was a correlation between online translation performance, and the "familiarity" of these stimuli. In particular, participants were asked to "rate the extent to which each novel object resembles a familiar object on a 5-point scale (i.e., does the novel object remind you of a particular known-object in any way?)", and stimuli that scored most highly were associated with better online translation invariance.

It is important to note however, that *online translation invariance was still strong for the*

least familiar stimuli, and robust online invariance for novel stimuli was also observed in Bowers et al. [2016]. Given our above findings, STDP learning alone may be limited in enabling spatial-configuration functions to contribute significantly to online translation invariance. An obvious possibility is that, as already noted, these computations could operate alongside pooling-based conjunction detection, and that the basis of online translation invariance in CNN architectures might have greater importance for the most novel stimuli. In other words, these computations are not mutually exclusive.

Alternatively, there is an obvious caveat to the conclusions from our simulations. In particular, and as in the work of Gerstner et al. [1996] and Izhikevich [2006], we have explored the sufficiency of STDP alone to enable the circuits of interest to emerge. As such, additional learning mechanisms in the brain may be able to offset the limitations we have noted above, thereby enabling spatial-configuration functions to still play a significant role for even the most novel stimuli.

In Bowers [2017] and Bowers and Davis [2017], it was suggested that myelin plasticity could serve as a means of directly modifying delay-lines, thereby inducing the required circuits to process spatiotemporal spike patterns like those proposed in Davis [2010b]. This would be a powerful basis of learning, adding an additional degree of freedom (cf the strength of a synaptic weight) to neural computations. Indeed, there is now strong evidence that myelin can be modified due to learning [McKenzie et al., 2014], and in an activity-dependent manner [Gibson et al., 2014, Hines et al., 2015, Bacmeister et al., 2021]. We have identified several limiting factors when our circuits relied on STDP alone, particularly if these computations are to be relevant for particularly novel stimuli. The interesting question then arises, which of these limitations might myelin plasticity serve to address, or at least mitigate?

While the probability of position functions sharing a linear bias, as well as the match probability discussed in Chapter 3, are independent of the conduction delays associated with axons, the alignment probability is naturally dependent on this. As such, myelin plasticity could enable connections from a common assembly (i.e. a match) to be adjusted such that signals arrive in a narrow temporal window. This would thereby limit a significant drop in the expected number of output neurons that can participate in spatially-sensitive conjunc-

tion detection. As such, a smaller sub-population of output neurons would be sufficient to achieve the robustness that we observe in our simulations after scaling to the largest population size (ensemble networks).

A related issue is that with large temporal offsets in the times at which neurons spike, alignments are unlikely to take place if these must emerge randomly from a distribution capped at e.g. 15ms. While longer delay-lines have been observed between more distant neural structures (e.g. up to 32ms between primary somatosensory and motor cortex [Swadlow, 1994]), myelin plasticity could address this issue by ensuring a sufficiently long conduction delay exists even when the temporal offset between features is relatively large. While this could unlock the use of larger temporal offsets between features (and thereby support larger integration windows), it is unclear how well the subsequent distribution of delay-lines would line up with the log-normal distribution observed in Egger et al. [2020]. As already noted, it may be that other distributions are observed in different parts of cortex and across various species.

In addition to the above effects concerning the probability of an alignment taking place, our simulation results demonstrated that the closer a post-delay-line alignment is to 0, the more robust a representation is to noise. While we observed that STDP has a preference for narrow alignments, sources of noise in the system, as well as the reality that many initial alignments will not be perfect, resulted in a wide variety of mixed alignments after learning. Myelin plasticity might once again prove beneficial by fine-tuning alignments, increasing the tolerance of output neurons to sources of noise from either jitter or diffusion in position functions.

In summary, myelin plasticity could enable the proposed computation to occur more frequently, and operate more robustly. In turn, this could result in such conjunction detection having considerable significance even for novel stimuli.

While the proposal that myelin plasticity could support the computations explored here is indeed an exciting one, we have previously noted that the precise learning rules that govern myelin plasticity in the brain are unfortunately yet to be well characterised [Pajevic et al., 2014, de Faria et al., 2021]. In particular, it is unclear if myelin plasticity can operate with sufficient anatomical and temporal precision to enable the fine-tuning that would be

desired. It may be, however, that even a coarse form of myelin plasticity could operate alongside STDP in a complementary self-organising scheme.

In particular, myelin plasticity could simply induce a distributional shift across a set of conduction delays (either slower or faster), after-which STDP could reinforce synapses associated with coherent delay-line alignments. STDP reinforcement of synapses might limit further myelin modification in associated axonal branches, while in circuits with limited alignments, myelin plasticity could continue as a soft search for a space able to support the desired alignments. Such a role for myelin in first being broadly applied, followed by activity-dependent maintenance of current myelin, is consistent with the experimental observations in Hines et al. [2015]. Similarly, recent evidence has suggested that myelin plasticity can operate in two phases, initially slowing, and then speeding up conduction in axons associated with a learned task [Bacmeister et al., 2021]. In summary, establishing how to implement a circuit that dynamically self-organises using biologically plausible forms of myelin plasticity remains an open, and interesting, avenue for future research.

4.6.3 Potential Limitations

The analyses we performed in Chapter 3, as well as our simulation results, were made tractable by our assumption that two nearly-coincident afferent spikes are sufficient to indicate the presence of an object. Our use of more biologically plausible conductance-based synapses complicates the ability to predict the integrative properties of neurons (the integration window), which we were able to offset by this assumption. As we previously noted, strong weights are observed biologically [Thomson et al., 1993, Lefort et al., 2009], where these may have particular significance for circuit structure [Song et al., 2005]. In our simulations, each neuron also received thousands of smaller Poisson inputs, which are necessary to bring the neuron to threshold (i.e. the coincident spikes are not sufficient), consistent with the view that integrative computations can operate alongside coincidence-detection. It is useful, however, to comment on the generalizability of our results if we were to scale up the number of input neurons per spatial-configuration assembly. This might occur either through multiple neurons representing each feature, and/or the addition

of more features.

Of greatest importance to our results is the integration window being sufficiently narrow to operate given a smaller ISI, following the addition of more input neurons. Fortunately, previous work using simpler neural models has demonstrated that decreasing the size of excitatory postsynaptic potentials also decreases the pressure on maintaining a large ISI, enabling networks to remain in a coincidence-sensitive regime [König et al., 1996, Rossant et al., 2011]. In other words, although more afferent neurons would be involved, scaling of their weights (such that e.g. 5 coincident inputs are required for an output spike) appears to be associated with similar integrative properties to the simpler setting of two-coincident inputs. This reinforces the notion that while our working definition of the integration window is clearly simplified, we were not only able to make several predictions about our own simulations, but in addition these predictions (such as to noise tolerance) should generalize as one scales to more complex, realistic spiking neural networks.

Another possible limitation of our model is that we did not account for the perceptual evidence of a feature's presence. More concretely, our simulations assume that the low-level features are unambiguously visualized, and the only computation that needs to be performed is determining their relative arrangement. In reality however, low-level features may only be partially visible, or otherwise obscured, such that there is uncertainty about their presence. How might this be accounted for in the model? In Davis [2010a], the amplitude of a letter signal encodes information about letter evidence, thereby disentangling this information from letter position. It was proposed that a neural level, this would be reflected in the number of feature neurons that become active in a population. Similarly, in capsule networks [Sabour et al., 2017], the evidence of a feature's presence is captured in the normalized length of the capsule vector, which is separate from the encoding of pose information in a vector's direction.

In Section 3.3 of Chapter 3, we discussed how perceptual evidence might be implemented biologically by the probability of an input neuron spiking, or the influence of firing rate inputs. The former is similar to the approach in Davis [2010a], where the probability of an input feature neuron's binary spike is a function of perpetual evidence. As such, the

number of output neurons that ultimately receive spikes would be affected by the perceptual evidence for each feature. A read-out mechanism that integrates over the amount of neurons activated in the output population would thereby be sensitive to the degree of perceptual evidence for an object.

The second mechanism we considered is that the Poisson firing rates associated with a given feature would change based on its perceptual evidence, where these signals would arrive via low-weight synapses. If the input firing rates are sufficiently high, then output neurons are able to integrate coincident inputs and fire. If, on the other hand, perceptual evidence decreases, then the membrane potential of output neurons would be too low, regardless of the temporal precision with which input signals arrive. Demonstrating the feasibility of this principle, the noisy background inputs in our network resulted in a similar effect. In particular, due to their activity, the membrane potential was at a noisy distribution centred around -72.5 mV, which is sufficient to reach the firing threshold if spikes from input neurons arrive within a narrow window. With lower background activity, however, it would be much harder (or potentially impossible) for output neurons to reach the spiking threshold, regardless of whether temporally coincident signals arrived from input neurons. As such, while not explicitly considered in our simulations, a variety of plausible mechanisms exist to account for feature evidence in the networks we have explored.

As a final limitation, it is worth addressing the question of encoding additional dimensions of variation. For example, we have considered translation invariance along one dimension, such as for vertical translation. Naturally however, objects can also move along the horizontal, and indeed there may be other dimensions of variation (such as rotation), for which we would wish to achieve some degree of invariance. In Davis [2010b], the authors modelled word recognition, for which a one-dimensional code is sufficient. In contrast, capsule networks [Sabour et al., 2017] are capable of modelling multiple degrees of variation in a single capsule. While different sets of spiking neurons could similarly be dedicated to feature conjunctions in different dimensions, it remains unclear how best to implement such functionality in the networks we have considered. As before, ensuring some degree of spatial sensitivity of feature detection (e.g. along one or two dimensions)

may be sufficient for object recognition that is more robust than current approaches.

4.6.4 On The Sufficiency of Pooling-Based Conjunction Detection

We have previously identified that, while it is clear that the pooling operation performed in CNNs will reduce the spatial precision with which features are encoded, and that this may have downstream consequences, the portrayal of such architectures as bag-of-features detectors is a significant over-simplification. In our work, we have focused on the processing that occurs within a single receptive field area, and thus the capabilities of a set of downstream neurons receiving inputs from invariant features. In a full architecture with receptive fields of mixed sizes and overlap, intuitions are more difficult to arrive at, and it remains an open question whether such computations are actually sufficient for robust conjunction detection, without the need for the computations that we consider. For example, while more classical CNN architectures such as VGG [Simonyan and Zisserman, 2015] can be approximated by bag-of-feature detectors, they are not *equivalent* to them [Brendel and Bethge, 2019]. Moreover, the addition of skip connections [He et al., 2016] or the concatenation of low-level feature maps [Huang et al., 2017] complicates intuitions about receptive fields and spatial processing even further. In particular, architectures with these modifications demonstrate unique properties such as less equivalence to bag-of-features processing [Brendel and Bethge, 2019] and in some instances [Huang et al., 2017], enhanced online translation invariance [Biscione and Bowers, 2021]. Finally, processing in CNNs has been observed to become more aligned with human vision following the introduction of appropriate [Geirhos et al., 2019, Blything et al., 2021] or sufficiently scaled [Geirhos et al., 2021] training data, suggesting this as a promising avenue for more faithful models.

Our work has been motivated by the observation that current machine vision models remain vulnerable to a variety of perturbations for which primates are robust, and that since their adoption, pooling operations have been noted as potentially problematic for robustness owing to the loss of spatial information [Hummel and Biederman, 1992, Serre et al., 2007]. It remains an open question whether the computations we have considered here are truly necessary to explain conjunction detection in primate vision, or whether current

approaches will be sufficient given other, more minimal modifications.

4.6.5 Experimental Predictions

While our results have served primarily as a feasibility study of the computations we explored, it is interesting to consider what experimental predictions our results might enable. The distribution of conduction delays we observe, while qualitatively similar, differs from that measured in Egger et al. [2020]. In particular, we suggested that spatial-configuration functions might have particular computational significance for more familiar stimuli. More generally, one would not expect the conjunction detection we consider to have much relevance to features that are largely independent of spatial structure. One interesting possibility would therefore be to measure the distribution of conduction delays observed in primate visual cortex, where one might observe a gradient of distributional changes as one moves between regions that show preferential processing to different stimulus categories. In particular, one would expect shorter, narrower conduction delay distributions in regions sensitive to unstructured textures, but as one moves to regions more sensitive to objects [Malach et al., 1995], and finally to those associated with expert forms of object recognition [Gauthier et al., 2000], one would expect a distribution of delays more similar to those that we have observed in our simulations.

Similarly, we have highlighted the relationship between conduction detection computations, the sparse assemblies we consider, and larger synaptic weights. If the computations we explored are fundamental to neurons that are sensitive to the spatial arrangement of features (such as the face and hand neurons observed by Desimone et al. [1984]), then measurements of their response profiles might identify an outsize role of a small number of large synapses in driving their activity. In contrast, we would expect that neurons with little sensitivity to the spatial arrangement of features would display more integrative properties, being driven almost exclusively by large numbers of weaker synapses.

Taking these two considerations together, if neurons can show configural sensitivity without the use of a small number of larger synapses, and in cortical regions without longer conduction delays, then it is unlikely that the computations we have explored are necessary for robust conjunction detection in the primate brain.

4.6.6 Conclusion

In spite of the impressive capabilities of machine learning models on complex visual tasks, and their ability to match considerable data in primate vision, the precise basis of conjunction detection remains unknown. In the two proceeding chapters, we have investigated the ability of spiking neural networks to implement a form of spatially sensitive conjunction detection, taking inspiration from machine learning models known as capsule networks [Sabour et al., 2017], and building on a computational neuroscience model of word recognition [Davis, 2010b]. We have demonstrated that spatial-configuration functions, along with STDP-mediated selection of delay-lines, could indeed represent a means by which the primate system achieves representations that are both robust to online translations, as well as sensitive to the configuration of a common set of features. We look forward to future work determining how these computations could operate alongside pooling-based conjunction detection, as well as the potential role for myelin plasticity in supporting the emergence of delay-line circuits.

5 | Hierarchical Binding in Convolutional Neural Networks: Making Adversarial Attacks Geometrically Challenging

5.1 Introduction

In this chapter, we approach the issue of robust machine vision by presenting a novel deep-learning architecture, inspired by work in theoretical neuroscience on how the primate brain performs hierarchical binding through polychrony. In particular, we explore the potential computational significance of the hierarchical binding representations discussed in Chapter 1, Section 1.2.4.2. We propose that the absence of such representations from current models might partly explain their vulnerability to adversarial examples, as it has been argued that such stimuli are a result of 'off-manifold' perturbations of images. Our novel architecture is designed to approximate hierarchical binding, providing explicit representations in these otherwise vulnerable directions. Having introduced these representations into convolutional neural networks, we provide empirical evidence of enhanced robustness against a broad range of L_0, L_2 and L_∞ attacks, particularly in the black-box setting. While we eventually report that the model remains vulnerable to a sufficiently powerful attacker (i.e. the defence can be broken), we demonstrate that our main results cannot be accounted for by trivial, false robustness (gradient masking). Analysis of the representational geometry of our architectures shows a positive relationship between hierarchical binding, expanded manifolds, and robustness. Through hyperparameter manipulation, we find evidence that robustness emerges through the preservation of general low-level information alongside more abstract features, rather than by capturing which specific low-level features drove the abstract representation. Finally, we propose how hierarchical binding relates to the observation that, under appropriate viewing conditions, humans show mild sensitivity to adversarial examples.

5.1.1 Motivation

Adversarial examples are images modified by small (L_p norm constrained) perturbations that cause machine vision systems to catastrophically misclassify objects [Szegedy et al., 2014]. Since their discovery, various efforts have been made at both explaining their existence, and conferring resistance to them [Goodfellow et al., 2015, Madry et al., 2018, Gilmer et al., 2018, Schott et al., 2019, Ilyas et al., 2019, Stutz et al., 2019]. This includes arguments that adversarial examples represent perturbations of the input off of the data manifold [Tanay and Griffin, 2016, Khoury and Hadfield-Menell, 2018, Stutz et al., 2019], and in particular that the vulnerability of a model is associated with the presence of many such directions being available to the attacker [Khoury and Hadfield-Menell, 2018].

Both biological and artificial means of object recognition typically propose that object classes can be effectively described by a lower-dimensional manifold than the input (e.g. pixel) space. Such a data manifold describes a region where samples are concentrated, and the dimensions of the manifold represent class-preserving changes to the object representation (such as increasing its size, or rotating it). Moving between object manifolds would generally correspond to low-density regions with potentially no associated object class, such as an image of white noise [DiCarlo and Cox, 2007, Bengio et al., 2013].

We argue that part of the phenomenon of adversarial examples is that architectural choices often assume object manifolds that are too low-dimensional. While this is sufficient for classification on a standard data-set, where samples off of the data manifold are sparse, many adversarial examples can exploit this assumption by moving orthogonal to the concentrated sub-space [Tanay and Griffin, 2016, Khoury and Hadfield-Menell, 2018, Stutz et al., 2019]. Given that humans are (in most cases by definition) robust, this implies that adversarial examples exist *on* the manifolds that underlie human classification of objects, and that these are therefore of higher dimension. One approach is therefore to augment the data manifold, such as by adding noise during training [Zantedeschi et al., 2017, Ford et al., 2019, Rusak et al., 2020], in order to more faithfully sample from the data manifold of images that a human would classify as a particular object. We argue however that typical convolutional neural network (CNN) architectures are constrained in their ability to practically model the spectrum of variation that an object can undergo. In particu-

lar, typical architectures result in the loss of low-level spatial details in order to promote invariance and linear separability of classes. We propose that a model that maintains standard classification abilities while modelling objects more faithfully in high-dimensional space should enable greater robustness. This representational expressiveness of a model can be measured as the extent of an object's manifold in the neural state space of the model, the neural manifold.

Assuming a higher-dimensional manifold underlies human perception, and that a model can capture this, many adversaries could be viewed as on-manifold adversarial examples, after which robustness is a case of standard generalization [Gilmer et al., 2018, Stutz et al., 2019]. We introduce a novel architecture that, through hierarchical binding, captures low-level features *alongside* lower-dimensional, invariant representations. The difficulty then is learning a useful decision boundary and sampling sufficiently in this high-dimensional space. For this reason, we complement our approach with regularization and data-augmentation with noise. The geometric intuition for our approach is summarized in Figure 5.1.

We note that our defence operates not by guaranteeing the absence of any vulnerable regions in the decision boundary, but by reducing the number of directions in which such regions are found. Indeed, in our Discussion we suggest that, given a sufficiently powerful attack method, the network is as vulnerable as a model without our defence (i.e. the defence can be broken). However, our improvements to the classifier's decision boundary provides robustness against a variety of methods for generating adversarial attacks, including several black-box methods that rely on these vulnerable directions being present, and which represent the primary methods that could be plausibly leveraged against humans.

Many proposed adversarial defences have shortcomings or can be entirely circumnavigated given an appropriate attack, a notorious issue in the literature that has grown increasingly apparent, and a reality affecting the vast majority of published defences [Tramèr et al., 2020b, Croce and Hein, 2020]. One of the only methods known to consistently improve robustness independent of the attack method is adversarial training [Madry et al., 2018], or variants of it [Gowal et al., 2020], although it can suffer from issues such as

over-fitting to particular distance metrics used in training [Schott et al., 2019, Laidlaw et al., 2021], and a loss in classification accuracy on clean, unaltered data [Tsipras et al., 2019].

Despite the seemingly intractable nature of adversarial examples, humans are, by definition, robust to them under normal conditions. Similar to the approach taken by recent defences on breaking certain attacks [Xiao et al., 2020], rather than defending against any possible attacker, we demonstrate a defence method with utility against a variety of black-box attacks, including naive brute-force attacks based on noise. Our defence is notable due to its a) connection to the growing geometric view of adversarial examples, b) ability to generalize across multiple distance metrics, a stumbling block for many adversarial defences [Schott et al., 2019, Laidlaw et al., 2021] and c) relevance to experimental results on the perception of adversarial examples in humans. Importantly, the connection we establish between our approach and the geometry of a neural network's representations lends the defence to future extensions that might enable models that are fundamentally robust to all attackers.

Our contributions are as follows:

- We present a mechanism to preserve and utilise information about which lower-level features play an important role in driving a CNN's higher-level, abstract representation at a succeeding layer. Our novel architecture is inspired by work predicting the existence of neurons in the primate brain that encode the hierarchical binding relationships between visual features at different spatial scales [Eguchi et al., 2018], and our findings offer a possible explanation for the mild sensitivity of humans to adversarial examples under appropriate viewing conditions.
- We present empirical results showing the robustness of these augmented networks to a variety of adversarial attacks, following the use of techniques to ensure a useful decision boundary. These results are complemented with findings from a brute-force noise attack that support the connection between the observed robustness and our geometric framing of the defence.
- We analyse the geometric properties of our networks' representations, finding that

the neural manifolds in hierarchical binding networks are of higher dimension and radius, quantified by the manifold extent. We further observe that, while neither necessary nor sufficient for resistance to adversarial attacks, greater neural manifold extent is associated with increased adversarial robustness.

- We additionally use iterative adjustments of a key hyperparameter (the γ-proportion) to demonstrate that the observed robustness is primarily a consequence of the low-level representations that hierarchical binding preserves, rather than the explicit encoding of binding relations. Hierarchical binding in the primate brain might thereby serve a variety of computational functions, with one of its benefits being robustness through the preservation of low-level information.

5.2 Related Literature

5.2.1 Hierarchical Binding

In this chapter, we investigate the possible significance of learned (i.e. statistically regular) hierarchical binding representations for the problem of adversarial examples. Hierarchical binding concerns how representations can capture the relations between features at multiple scales of abstraction, in particular the relationship between a low-level feature and a high-level feature that it causally contributes to [Treisman, 1996, 1999, Eguchi et al., 2018]. For example, when we look at an alphabetical letter T, we can see the vertical and horizontal bars that comprise the letter as distinct elements, as well as the fact that these constituent elements are part of the letter T itself. This also entails an understanding that a particular vertical bar feature is part of the T and not e.g. a letter L nearby. Low-level features carry meaningful spatial information due to their small receptive fields and high dimension, and hierarchical binding can thereby encode class-preserving transformations of an object - note in our example that it is not simply the abstract concept of a vertical bar that is bound to the T, but the specific representation localised in space. We emphasize that this framing does not discount the importance of invariant features; rather the goal is to capture a continuum of abstraction jointly.

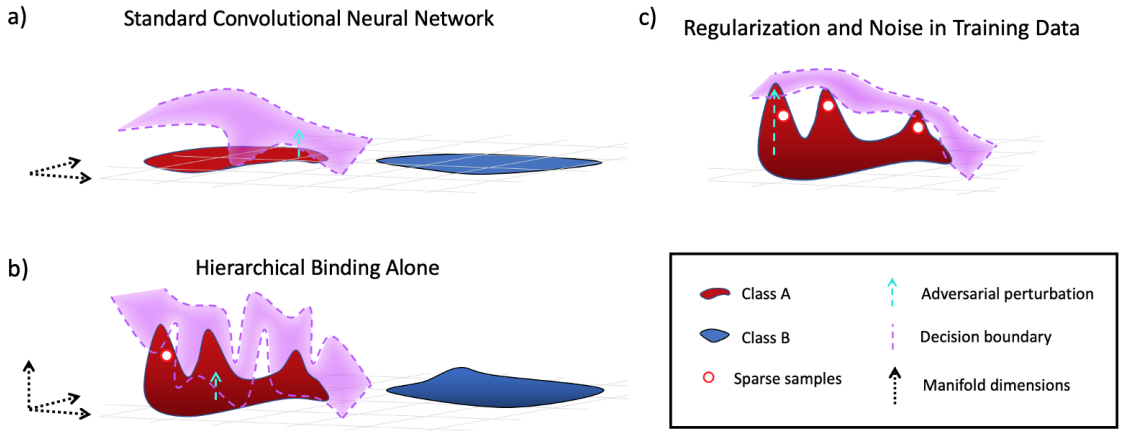


Figure 5.1: *The effect of hierarchical binding on the decision boundary.* Red and blue represent two different object manifolds (e.g. cats and dogs). Adversarial perturbations (light-blue arrows) for the red class move the input beyond the decision boundary into a region where it is classified as blue. a) A common approach for classification is to assume low-dimensional representations of objects are sufficient, as these support linear decision boundaries that accurately separate the objects. Unfortunately, the learned decision boundary can be unpredictable off the manifold. Given the high-dimensional embedding space (e.g. pixel-space), there may be many such directions vulnerable to small perturbations. Here, an idealized 2D manifold is used to depict the data distribution typically assumed when designing model architectures. b) We argue that there are additional, class-preserving dimensions of variation to the manifold, where the shown manifold represents all examples a human would classify as a given object (here depicted as a 3D solid). We suggest that these dimensions are impractical to model with typical convolutional neural network (CNN) architectures. Adding hierarchical binding enables the network to explicitly represent these features alongside the more abstract dimensions, potentially aligning the representation more closely with the basis of human perception. Despite this modification, the sparsity of samples in high dimensions (particularly in standard data-sets) means that further steps are required for a robust decision boundary. c) Using noise during training helps sample more densely from the manifold underlying human perception, which alongside regularization, provides a more robust decision boundary against a variety of adversarial attacks.

Eguchi et al. [2018] proposed a mechanism by which the brain might capture hierarchical binding, encoding the relations between low-level and high-level features throughout the visual processing stream. This work predicted that temporal coincidence detection, conduction delays, and lateral and top-down connectivity would all be important for implementing this binding mechanism. In particular, these properties could enable spiking neurons or assemblies of spiking neurons to fire if and only if a particular low-level feature was causally driving a higher-level representation. For a more in-depth summary of hierarchical binding through such temporal coding (polychrony), please refer again to Section 1.2.4.2 in the main introduction (Chapter 1). Rather than simulate these biological elements, we use a non-local algorithm to approximate such representations at the computational level [Marr, 1982] within an adapted CNN architecture. We also use a simplified form of the full hierarchical binding formulation proposed in Eguchi et al. [2018] in order to constrain the computational cost of the architecture, enabling us to explore the significance of the proposed representations for robust object classification. We emphasize that while previous work has investigated the emergence of hierarchical binding in spiking neural networks [Eguchi et al., 2018, Isbister et al., 2018], this work did not assess its relevance to adversarial robustness.

It is clear that many aspects of primate vision rely primarily on abstract, low-dimensional representations, which is consistent with our proposal here, and that vision can be impoverished e.g. outside of attention. What we are describing here however is the primate perception of "objects along with their detailed features" [Lu et al., 2018], and indeed perceptual processing in primates accesses such low/mid-level information [Kim and Feldman, 2009, Kang and Maunsell, 2020]. In other words, the representations we explore are consistent with "vision with scrutiny", rather than coarse object representations [Hochstein and Ahissar, 2002].

5.2.2 Preserving Low-Level Information in Computer Vision

Many methods exist to preserve low-level information in deep neural networks [Srivastava et al., 2015, Ronneberger et al., 2015, He et al., 2016, Huang et al., 2017, Jacobsen et al., 2018], spatially enrich feature representations [Sabour et al., 2017] or encourage

the encoding of additional factors of variation [Cheung et al., 2015]. Our architecture is novel in that it captures which low-level neurons causally drove the network's high-level representation, and explicitly encodes such information as layers in their own right for classification. This explicit encoding is important for downstream read-out of the representations. In contrast, architectures using skip connections typically combine information from low and high-level layers in an additive operation [He et al., 2016]; this can obscure the respective contributions of features and makes classifier read-out of the low-level details more challenging. Furthermore, our architecture specifically combines low-level features with a high spatial resolution (i.e. cross-wise dimension) alongside more abstract representations with coarser spatial information. Where prior architectures concatenate features, such a preservation of spatially detailed and coarse features is not performed [Huang et al., 2017]. In summary, our approach is related to the motivation for (and biological evidence of) disentangled representations [Higgins et al., 2021]. In particular, our architecture is biased so as to "disentangle as many factors as possible, discarding as little information about the data as is practical" [Bengio et al., 2013]. This motivation is important as disentangled representations are associated with more sample-efficient learning [van Steenkiste et al., 2019], while learning a robust decision boundary is a sample inefficient problem [Gilmer et al., 2018, Khoury and Hadfield-Menell, 2018].

5.2.3 Adversarial Examples

While there is a large literature on adversarial examples (see e.g. Yuan et al. [2019] for a review), we focus on those papers that are most relevant to the current work. The concept of adversarial examples as manifold failures has inspired several defences [Jalal et al., 2017, Samangouei et al., 2018, Song et al., 2018, Schott et al., 2019, Jang et al., 2020]. Stutz et al. [2019] showed that typical adversarial examples move orthogonal to the data's low-dimensional manifold, and Khoury and Hadfield-Menell [2018] provided evidence in synthetic data-sets that a greater number of directions normal to the data manifold is associated with increasing vulnerability. This appears to be because the decision boundary can be arbitrary off of the manifold [Khoury and Hadfield-Menell, 2018], and may indeed lie very close to it [Tanay and Griffin, 2016]. Work that followed our own has recently shown that the stochastic nature of neurons in the primate visual stream is associated

with a more stable manifold representation in the face of adversarial examples [Dapello et al., 2021]. Minor results in this study are largely consistent with the results that we will present, specifically the relationship between manifold extent, and adversarial robustness. Unlike Dapello et al. [2021], we present a novel architecture explicitly designed to better model off-manifold representations, and demonstrate its enhanced robustness on a variety of challenging black-box attacks. Furthermore, we highlight that while we introduce additive noise at training time, the behaviour of our network is deterministic. This is in contrast to the model in Dapello et al. [2021], where they were specifically concerned with the significance of stochastic neural behaviour, including at inference time. This distinction is important, as we are will consider the contribution of noise at training time to a more robust decision boundary.

We also note that previous work on extracting hierarchical interpretations for the predictions of neural networks has shown that these interpretations themselves can be resistant to adversarial attacks [Singh et al., 2019], although this work did not relate to the issue of robust object classification.

In our Discussion (Section 5.8), we address the relationship of our model to other biologically-inspired defence mechanisms.

5.3 Model Description

5.3.1 Approximating Hierarchical Binding

We use as the basis for our architecture convolutional neural networks (CNNs). Figure 5.2a includes a demonstration of what a typical convolution and max-pooling operation might look like. Figure 5.3a shows how a typical CNN architecture implements successive convolution and max-pooling operations, alongside our additional operations of unpooling and gradient unpooling (introduced below). While the successive use of convolutions and max-pooling are not without their limitations (as discussed elsewhere), they capture the key principles that are believed to operate in the primate brain - that is feature detection and invariance.

To capture which low-level features causally drove max-pooled representations, we use the operation known as unpooling. Max-pooling determines the maximum value of each type of feature over a local spatial region as a way of achieving a location invariant representation of that feature. However, such an operation loses potentially useful information about the location of that feature within the spatial region. Unpooling is used here to recover and preserve such location information, by the addition of a sub-layer of unpooled feature activations. The unpooling operation identifies which locations had maximal activations (i.e. survived max-pooling), and preserves these activations at their locations in the new sub-layer of unpooled feature activations, while the activations for that feature which did not take maximal values are set to zero [Zeiler and Fergus, 2014, Badrinarayanan et al., 2017]. This procedure is illustrated in the top right of Figure 5.2a. A modified version of unpooling, termed 'ratio unpooling', has previously been employed in a mixed bottom-up and top-down network as a means of preserving spatial information [Xu et al., 2019]. Importantly however, this modified form of unpooling does not capture what lower-level features causally drove higher-level features (i.e. hierarchical binding), consistent with this not being the motivation of the authors.

To capture which simple features contributed to abstract representations, we introduce what we term 'gradient unpooling' (Figure 5.2a). Capturing how every feature in a lower layer (denoted L^k) contributes to a feature in a succeeding layer (L^{k+1}) preserves full information about these hierarchical binding relations [Eguchi et al., 2018], but is computationally unappealing due to the number of required binding representations. This is particularly problematic if this information is fed into a fully-connected layer for learning decision boundaries in a classifier. As such, we propose a computationally more feasible approach where we treat the high-level representation as distributed, and assume that only one object is ever represented in this distributed activity at a time (a valid assumption for the data-sets we use). Our aim is then to capture what low-level features contributed to this distributed representation.

To do so, we assume that the partial derivative of a high-level activation within a hierarchical network, taken w.r.t. to a low-level activation, carries some information about the significance of the contribution of that low-level feature. This is therefore our primary

signal for causal influence, and this principle of using gradients is an established approach to attributing the significance of a feature, for example where the partial derivative of a class score is taken w.r.t. pixel inputs [Simonyan et al., 2014]. As we are interested in the contribution of the low-level feature neuron to the distributed representation in the higher layer, we use the sum of the partial derivatives across the high-level layer. Finally, as the scalar value of a low-level neuron's activation provides information about its presence in the image (and therefore its likelihood of having contributed to higher-level representations), we include this in the final gradient unpooling values.

We now describe gradient unpooling in detail. Our first aim is to use the proposed gradients to approximate the significance of a low-level unit a_i with the measure m_i. Specifically let $a_i^{(k)}$ be the activation of an ith unit in the lower-level layer $L^{(k)}$, and $a_j^{(k+n)}$ in the higher-level layer $L^{(k+n)}$, where $n \geq 1$. The measure units m_i form a tensor of the same dimension as $L^{(k)}$, where each unit's value is given by:

$$m_i = \sum_j \frac{\partial a_j^{(k+n)}}{\partial a_i^{(k)}} \tag{5.1}$$

This sum is taken over the activations $a_j^{(k+n)}$ in $L^{(k+n)}$. As the gradient is taken over the entire higher-level layer, a proportion γ of the largest gradients are then selected with the intent of capturing the most important driving neurons. Here γ is a hyperparameter between 0 and 1. The winning gradients are used to generate a Boolean mask applied to the activations of the low-level layer. Denoting our approximation of a layer k unit's causal role with $c_i^{(k)}$, the final sub-layer of gradient-unpooling values are then given by

$$c_i^{(k)} = \begin{cases} a_i^{(k)}, \text{where } m_i \in \text{top-proportion}_\gamma(m) \\ 0, \text{otherwise} \end{cases} \tag{5.2}$$

The sub-layer $c^{(k)}$ is then up-projected along-side the unpooling sub-layer. Intuitively, a neuron might have a large activation, but if many of the high-level neurons with which it shares a large weight did not survive max-pooling (i.e. it did not successfully drive them), then its summed gradient m_i is likely to be smaller than that of other neurons. The unit

is thus unlikely to survive the γ cut-off and have its activation preserved in the gradient unpooling sub-layer. On the other hand, a neuron with large learned weights to the higher-level features that survived max-pooling is more likely to survive the γ cut-off. Up-projecting the activation value itself then provides some additional measure of how likely the low-level neuron was to have genuinely contributed to the abstract representation in the higher layer.

Our approach seeks to capture the low-level neurons that contributed to the distributed representation in the higher layer, although it is only an approximation of the actual causal relations between them. We also explored alternative mathematical formulations, such as including the activity of the low-level neuron in deriving the Boolean mask. That is, using

$$m_i = a_i^{(k)} \times \sum_j \frac{\partial a_j^{(k+n)}}{\partial a_i^{(k)}} \tag{5.3}$$

in place of equation 5.1. This offered comparable performance, but was more difficult to implement in the libraries we leveraged, and therefore equation 5.1 was used. Better measures of the importance of a low-level neuron to a higher-level representation exist (see e.g. Dhamdhere et al. [2019]), but we use our method due to its computational efficiency [Simonyan et al., 2014]. In our networks, we concatenate the results of unpooling and gradient-unpooling alongside the max-pooled activations in the feed-forward stream. This serves to provide both invariant and spatially detailed representations jointly.

In Figure 5.2b, we show how these operations relate to hierarchical binding, as described in Eguchi et al. [2018] and Isbister et al. [2018]. Note that while these prior works proposed capturing every binding relationship between successive layers k and $k + n$, we seek only to preserve information about how low-level neurons have participated in driving distributed representations in the higher layer. As we demonstrate later, this additional information appears sufficient to enable a more robust object representation when targeted by adversarial attacks.

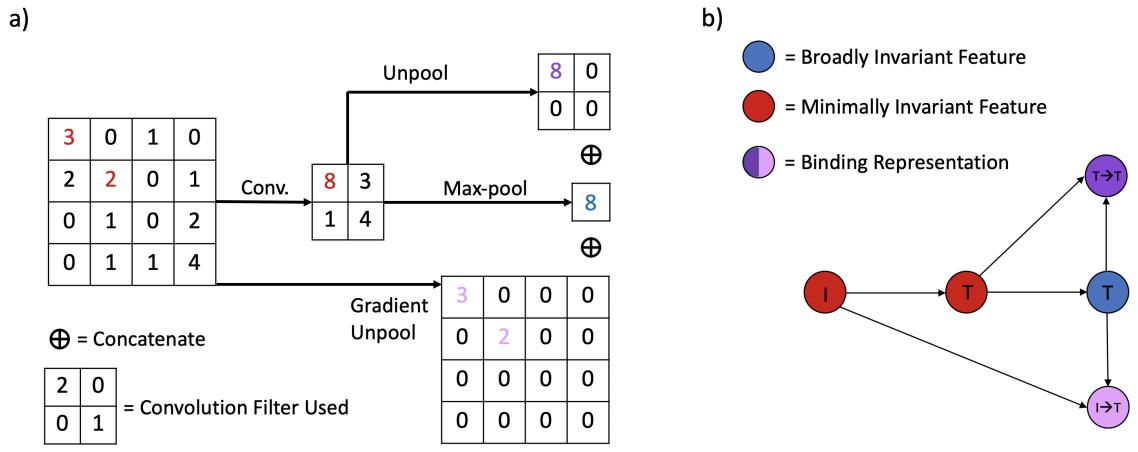


Figure 5.2: *Implementing hierarchical binding in a convolutional neural network.* a) 'Conv.' is a convolution operation with stride of 2 (that is, the weight matrix is shifted by 2 units after every convolution). Our depicted representation of gradient unpooling is simplified for the sake of intuition, as the max-pooling layer in the figure consists of only a single neuron; in reality, we take the gradient of each low-level activation w.r.t. the entire max-pooled layer and only use the proportion γ of the largest gradients to apply a Boolean mask to the activations. b) A toy diagram to demonstrate the connection to hierarchical binding. The desire is to capture which low-level features (such as a vertical bar or a minimally invariant 'T' neuron) causally drove the more invariant representation of a 'T'. Arrows in (b) indicate that a node participated in activating a representation, rather than simply the existence of a connection. These hierarchical binding representations are then made available, alongside the invariant representations, to higher layers.

5.3.2 Model Architectures

Both the unpooling and gradient unpooling computations can be introduced into standard CNN architectures, potentially at multiple levels. This work covers its use for CNNs used on the MNIST [Lecun et al., 1998], Fashion-MNIST (FMNIST) [Xiao et al., 2017], and CIFAR-10 [Krizhevsky, 2009] data-sets, which consist of hand-written digits, grey-scale images of clothing, and colour images of objects such as horses and airplanes respectively. The CNNs used for MNIST and FMNIST are based on the LeNet-5 architecture [Lecun et al., 1998]. In our Hierarchical Binding CNN (HBCNN), the LeNet-5 architecture is augmented with one unpooling and one gradient unpooling sub-layer (Figure 5.3a). The models used for CIFAR-10 are based on a deeper CNN using a VGG-like architecture [Simonyan and Zisserman, 2015] (Figure 5.3b). For each architectural variant (including the control models without hierarchical binding), hyperparameter tuning for adversarial robustness was performed on a cross-validation data-set (10k examples held out from the original training data-set). To provide an unbiased measure of the effect on robustness, 30 randomly generated networks for each architectural variant were then trained on the full training data-set and evaluated on the test data-set, with the median performance reported in all following results. Further model details are provided in 5.3.4.

5.3.3 Regularization

Regularization is a means of preventing a network from over-fitting data with a highly complex decision boundary, as this can generalize poorly to data not seen during training. To perform regularization, we introduce label-smoothing to some of our models, a method that reduces over-confident predictions by replacing the typical one-hot label vector (where the ground-truth class label is associated with a probability of 1, and all other class labels a probability of 0) with a 'one-warm' vector [Szegedy et al., 2016, Pereyra et al., 2019]. Specifically, consider the probability distribution of the ground truth labels for a class y where $p(y) = 1$ and $p(k) = 0$ for all other classes $k \neq y$. To perform label smoothing, a smoothing parameter δ and the number of classes K are used to assign the

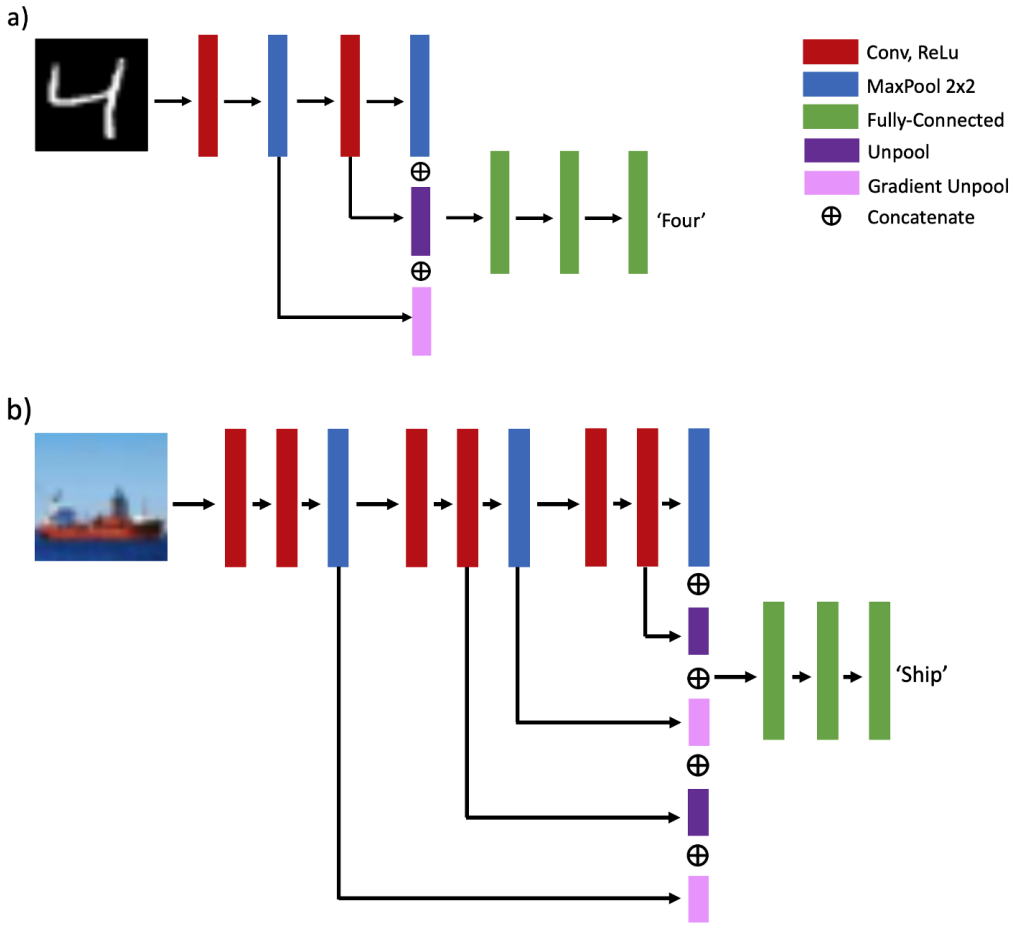


Figure 5.3: *Integration of Unpooling and Gradient Unpooling in a Hierarchical Binding CNN.* The diagram shows the architectures used for a) MNIST and Fashion-MNIST, based on the LeNet-5 CNN and b) CIFAR-10, based on a VGG-style CNN. Note that as it is not necessary to use unpooling or gradient-unpooling representations from every layer, deeper architectures can select intermittent representations to use for unpooling or gradient-unpooling, avoiding an excessive growth in parameters.

probabilities for incorrect classes as

$$p(k \neq y) = \frac{\delta}{K} \tag{5.4}$$

The target probability of the correct label is assigned as

$$p(y) = 1 - \delta + \frac{\delta}{K} \tag{5.5}$$

Thus if the label smoothing parameter was set to 0.1 (the value we use in all our models), the distribution over a total of 10 labels for the first class would change from $[1.0, 0.0, ..., 0.0]$ to $[0.91, 0.01, ..., 0.01]$. Label smoothing therefore models the concept that the data labels may contain errors, and that the system should be discouraged from creating an overly complex decision boundary to classify these [Goodfellow et al., 2016].

We use label smoothing as it is a widely implemented regularization technique for enhancing adversarial robustness [Warde-Farley and Goodfellow, 2016, Summers and Dinneen, 2019, Shafahi et al., 2019, Chen et al., 2021a, Pang et al., 2021]. While recent research [Lee et al., 2021] has identified that it can, under certain circumstances, contribute to gradient masking (a phenomenon we define below in Section 5.4.1), we take several steps to examine for the prevalence of gradient masking (as advised in e.g. [Pang et al., 2021]), highlighting it wherever any evidence exists.

With the exception of the 'vanilla' model described later, we also use weight decay in the VGG models to further regularize the decision boundaries. This form of regularization punishes large weights in the network by adding a penalty term to the cost function of the classifier based on the sum of the squares of the network's weights [Ng, 2004]. Details on the use of weight decay in the VGG networks is provided in 5.3.4.

5.3.4 Additional Model Details

All models are implemented in TensorFlow 1.14 [Abadi et al., 2016], and used the Rectified Linear Unit (ReLU) activation function [Nair and Hinton, 2010].

The LeNet-5 model used as the vanilla and label smoothing control for MNIST consists of

two convolution (6 and 16 channels, kernel size (5, 5)) and two max-pooling layers, followed by two fully connected layers of size 120 and 84. This baseline model was initially chosen as the starting point for our HBCNN, as the limited number of channels and use of two (rather than one) fully connected layers would help limit the increase in parameter numbers as binding was introduced. When we introduce noise in training, we use two fully connected layers of dimensions 256 and 128, as this offered better performance for both the LeNet-5 and HBCNN (LeNet-5 variant), consistent with this representing a more complex data-set to fit. For the HBCNN (LeNet-5 variant), the unpooling layer is applied to the last max-pooling layer, while the gradient-unpooling sub-layer is between the last max-pooling layer and the pre-convolution activations proceeding it (Figure 5.3a). For FMNIST, we used the same architectures as for MNIST with the larger fully connected layers.

We also evaluate two non-binding CNNs for MNIST with an equivalent number of free parameters to our HBCNN (LeNet-5 variant). The first of these has one fully connected layer, as this is a common architectural choice in adversarial example research, such as the robust model in Madry et al. [2018], and the baseline models in Schott et al. [2019] and Nassar et al. [2020]. This size-controlled model has two convolutions with channel sizes 32 and 64, and a fully connected layer of dimension 512, for a combined 877,440 parameters vs 850,070 in the HBCNN (LeNet-5 variant) - see Table 5.1. The second size-controlled model has two fully connected layers, as in the case of the LeNet-5 and HBCNN (LeNet-5 variant) models. It has two convolutions with channel sizes 54 and 108, and fully connected layers of dimensions 256 and 128, for a combined 872,954 parameters.

The two other architectures included in our MNIST results are the baseline CNN used in Nassar et al. [2020], and the adversarially trained CNN from Madry et al. [2018]. The former consists of two convolution (16 and 32 channels, kernel size (3,3)) and two max-pooling layers, followed by a single fully connected layer of dimension 1000. When training this network, we use the same hyperparameters as outlined in Nassar et al. [2020]. The adversarially trained CNN consists of two convolution (32 and 64 channels, kernel size (5,5)) and two max-pooling layers, followed by a single fully connected layer of

dimension 1024. Unlike the main CNNs leveraged in MNIST, note that this architecture uses 'same' padding for both convolutions, resulting in a larger (7×7 rather than 5×5) cross-section when feeding into the fully connected layer. This contributes to the much larger number of parameters (Table 5.1).

For the VGG architectures leveraged on CIFAR-10 (Figure 5.3b), we used three blocks, each containing two convolutions and one max-pooling (channel dimensions 32, 32, 64, 64, 128, 128, kernel sizes (3,3) throughout), followed by two fully connected layers of dimension 120 and 84, so as to once again constrain the growth in free parameters following the introduction of binding. For the HBCNN (VGG variant) we used two un-pooling sub-layers (corresponding to the 2nd and 3rd blocks), and two gradient unpooling sub-layers (from the 3rd max-pooling layer to the 1st and 2nd max-pooling layers respectively) (Figure 5.3b).

All of our models were regularized with dropout [Srivastava et al., 2014] of 0.25 and, where used, label smoothing of 0.1. Training of a network's parameters was performed using the Adam optimizer [Kingma and Ba, 2015] with a learning rate of 0.001 (MNIST and FMNIST) and 0.0005 (CIFAR-10), and a batch-size of 128. Training was performed for 45 epochs (vanilla MNIST model, and LeNet-based models with smoothing), 90 epochs (all other LeNet-based models, except for the 2-fully connected layers, size-controlled CNN, which showed considerable improvement on cross-validation data when trained for 180 epochs), and 500 epochs (all VGG models). For VGG models, we also augmented the training data with random shifts and horizontal flipping of the image.

For VGG models with smoothing and noise in the training data, we used L_2 regularization of 10^{-3} (standard) or 10^{-5} (HBCNN) for the weights from the final max-pooled layer, and 10^{-3} for the unpooling and gradient unpooling representations.

Our evaluation includes adversarially trained models for MNIST and CIFAR-10 loaded from the MadryLab Challenge repositories (`https://github.com/MadryLab/mnist_challenge`) and (`https://github.com/MadryLab/cifar10_challenge`), the latter of which is based on a Wide ResNet architecture [Zagoruyko and Komodakis, 2016]. These were generated by the original authors of Madry et al. [2018], and use adversarial training with the Projected Gradient Descent (PGD) attack described therein. An adversarially

trained model for FMNIST was loaded from the repository from Croce et al. [2020]); this model was trained using PGD attacks (40 iterations), with 50% adversarial images, and 50% clean images in each batch, for 100 epochs.

5.4 Evaluation Methods

5.4.1 Adversarial Attacks

Adversarial attacks are a means of producing images that are misclassified by machine vision systems, with the aim that the new image is as close to the original as possible. As measuring how large the modification is perceived to be by a human is challenging, optimization methods are generally used to create an adversarial image that minimizes a measured distance between the original image and the adversary. This distance is typically quantified by the L_0, L_2 or L_∞ norm. In one threat setting, the attack method has extensive access to the model being attacked (such as the gradients of its outputs w.r.t the pixels of the input), in which case it is termed a 'white-box attack'. In settings where the optimization process has limited knowledge about the model being attacked (such as only being able to query how it classifies any given image), the attack is known as a 'black-box' method [Brendel et al., 2018].

Inspired by the thorough evaluation of adversarial robustness in Schott et al. [2019], we evaluate our model against a broad range of black-box and white-box methods, covering L_0, L_2 and L_∞ norm measured attacks, described below. All attacks were evaluated using FoolBox v2.4 [Rauber et al., 2017]. As in Schott et al. [2019], our main result is the median distance of adversaries, as this is less affected by outliers than the mean, and unlike when reporting accuracy, is not vulnerable to over-fitting on an arbitrary threshold. For completeness, we also report the accuracy against adversaries bounded by a particular perturbation threshold ϵ. All results presented are based on adversaries generated from a subset of 512 images from the test data-sets. Below, we describe the main attacks used and their core intuition; in 5.4.1.4, we provide additional details such as hyperparameters for the attacks used.

Table 5.1: Number of Trainable Parameters in the Primary Models

	# of Parameters	**# input/output**
CNN (LeNet-5 variant)		
1st Conv.	156	1/6 c
2nd Conv.	2,416	6/16 c
1st FC	102,656	400/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**139,414**	
HBCNN (LeNet-5 variant)		
1st Conv.	156	1/6 c
2nd Conv.	2,416	6/16 c
1st FC	813,312	400+1600+1176/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**850,070**	
Size-Controlled CNN (1 FC)		
1st Conv.	832	1/32 c
2nd Conv.	51,264	32/64 c
1st FC	819,712	1600/512 u
Output	5,632	512/10 u
Total	**877,440**	
Size-Controlled CNN (2 FC)		
1st Conv.	1,404	1/54 c
2nd Conv.	145,908	54/108 c
1st FC	691,456	2700/256 u
2nd FC	32,896	256/128 u
Output	1,290	128/10 u
Total	**872,954**	
CNN+AT from Madry et al. [2018]		
1st Conv.	832	1/32 c
2nd Conv.	51,264	32/64 c
1st FC	3,212,288	3136/1024 u
Output	11,264	1024/10 u
Total	**3,275,648**	

Summary of the number of parameters that can be optimized during learning for the primary models leveraged on MNIST. Abbreviations: Conv. = convolution; FC = fully connected; c = channels; u = units; AT = adversarial training.

5.4.1.1 Gradient-Based Attacks

In typical training, one performs gradient descent of the loss w.r.t to the model's weights, where the loss is the term that measures the mismatch of the model's predictions to the ground-truth labels. Intuitively, a basic gradient-based attack can use knowledge of the model to perform gradient *ascent* of the loss with respect to the input *pixels*. As a result, pixels can be modified so as to cause the model to misclassify inputs, with the assumption that the modification to the pixels (if sufficiently small) will not have changed the ground-truth label as determined by a human. The Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015] uses this principle to construct adversarial examples as follows. Let $J(\theta, \mathbf{x}, y)$ be the cost function used to train the classifier with parameters θ, input $\mathbf{x}$, and targets y. Defining $\boldsymbol{\eta}$ as the adversarial perturbation added to an image, and scaling with parameter ϵ, then one can generate an adversarial perturbation as follows:

$$\boldsymbol{\eta} = \epsilon(\text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))) \tag{5.6}$$

where the adversarial image is then provided by

$$\tilde{\mathbf{x}} = \boldsymbol{\eta} + \mathbf{x} \tag{5.7}$$

Given the privileged access to the model's gradients these methods have, these are a form of white-box attacks. Various methods have been developed within this class that can be used to minimize either the L_2 or L_∞ norm of the adversarial perturbation; as in the Schott et al. [2019] evaluation protocol, we use the Fast Gradient Method (FGM), FGSM [Goodfellow et al., 2015], L_2 and L_∞ Basic Iterative Method (BIM) [Kurakin et al., 2019], L_2 and L_∞ DeepFool [Moosavi-Dezfooli et al., 2016], and Momentum Iterative Method (MIM) [Dong et al., 2018]. For MNIST and FMNIST, these attacks are also repeated with numerically estimated gradients, using the same method leveraged in Schott et al. [2019]. Unlike in their study, we also include Projected Gradient Descent (PGD, closely related to BIM) for MNIST, using multiple random starts [Madry et al., 2018].

5.4.1.2 Decision-Based Attacks

These rely only on the decision output of the network, and are therefore a form of black-box attack. A particularly powerful method is the Boundary Attack [Brendel et al., 2018]; intuitively, an image is first perturbed by noise until it is misclassified, after which the Boundary Attack iteratively moves the adversary closer to the original image while ensuring it remains misclassified. By taking small steps, it can treat the decision boundary as approximately linear and move along it. Note that the Boundary Attack therefore requires that at a sufficiently small step size, the decision boundary behaves linearly. If this does not hold, then it can fail to operate as intended, which we later discuss as an issue when it is leveraged against our own HBCNN model.

Another decision-based attack is the Pointwise attack. This method adds salt-and-pepper noise until an image is misclassified, then returns as many pixels to their original values as possible, whilst ensuring the image remains misclassified [Schott et al., 2019].

Finally, one can simply add increasing levels of noise (such as salt-and-pepper, uniform, or Gaussian noise) until an image is misclassified. As in Schott et al. [2019], we include the Boundary Attack, their L_0 and L_2 Pointwise Attack, the Salt&Pepper Noise attack and the Gaussian Noise attack. We also include the Blended Uniform noise attack, which blends the image with uniform noise until the input is misclassified [Rauber et al., 2017].

5.4.1.3 Transfer Attacks

Transfer attacks leverage the empirical observation that adversaries generated for one network (called the 'surrogate') can often transfer to other networks, even those with different architectures. The intuition is that the decision boundaries of different systems trained to perform the same task often align, and thus share vulnerable decision regions [Papernot et al., 2016, Tramèr et al., 2017]. Robustness to transfer attacks as well as decision-based attacks provides evidence that a model's robustness is not simply a result of 'gradient masking'. In gradient masking, some aspect of the network makes the gradients inaccessible or otherwise less useful for carrying out the optimization steps performed by gradient-based attacks. Often unintentionally introduced by architectural modifications, gradient masking is a common but less desirable method of adversarial robustness, given

its ineffectiveness against alternative methods of generating adversaries, and thus the false sense of security it provides [Papernot et al., 2017, Athalye et al., 2018].

In keeping with the above, transfer attacks are most often used to determine whether a particular model displays gradient masking. The presence of such gradient masking can result in the telltale finding that transfer attacks are more effective than white-box methods. However, transfer attacks are weaker than white-box attacks when no gradient masking exists, and they are less effective when it is difficult to craft attacks on the surrogate models used to generate the transfer attacks. They thus do not represent a gold-standard measure for a model's absolute robustness [Tramèr et al., 2020b].

In spite of this, robustness to transfer attacks represents its own interesting research question [Tramèr et al., 2017]. In particular, our defence is proposed to operate by reducing the number of directions in which the model has sub-optimal decision boundaries, and in which it is therefore vulnerable to adversaries. Transfer attacks exploit the shared dimensions of models with vulnerable decision boundaries [Tramèr et al., 2017], and as such, robustness to transfer attacks would support the geometric motivation behind our defence. Furthermore, and as we elaborate in our discussion, transfer attacks are the only method that has been leveraged against humans, and therefore the nature of models robust to transfer attacks (even when vulnerable to white-box attacks), is relevant to understanding the settings under which humans may display sensitivity to adversarial examples. For this reason, transfer attacks form an emphasis of our results, and we take steps to leverage a particularly broad transfer attack designed to identify multiple adversarial directions.

To develop this broad transfer attack, our main addition to the assessment protocol in Schott et al. [2019] is that we derive transfer attacks from multiple surrogate networks, including ones with our proposed architecture. This is to ensure that transfer attack robustness is not simply a result of the HBCNN having an exotic architecture, or being difficult to generate transfers from [Tramèr et al., 2020b], while still being fundamentally vulnerable. Specifically, adversaries are generated using the FGSM, L_∞ BIM, FGM, and L_2 BIM attacks. For FMNIST and CIFAR-10, these are leveraged against both a standard surrogate network and one augmented with binding. For MNIST, we use a particularly sophisticated transfer attack, where we use surrogates that represent multiple

versions of the target architecture, as well as the architecture it is being compared to for robustness. For each one of these variants, we use two independent surrogates, such that for any given target architecture, we generate transfer attacks from eight surrogate models. In addition to these, we generate transfer attacks from a ninth surrogate model that has been adversarially trained (note this is a different model from the one we evaluate later), as adversarially trained networks can often produce powerful transfer attacks [Carlini et al., 2019]. When attacking a network, all of these adversarial candidates (18 per distance metric) are leveraged. Using the same line-search from Schott et al. [2019], images are iteratively perturbed from the baseline image until they are misclassified, and the minimally perturbed, successfully adversarial transfer image is used in all distance and accuracy measures. Further details on our surrogate models are provided in 5.4.1.4.

5.4.1.4 Hyperparameters for Adversarial Attacks

For PGD we used 20 random starting points for the adversary, followed by 250 iterations with an initial step-size of 0.01, selecting the best adversary of the 20 for each image. For BIM we used 10 iterations with an initial step-size of 0.05. MIM was applied with 10 iterations, an initial step-size of 0.06, and a decay factor of 1.0. For PGD, BIM, and MIM, the step-size and epsilon were automatically adapted in Foolbox using a binary search. For MNIST, we combine 10 runs of the Pointwise L-0 attack, as performed in Schott et al. [2019], taking the minimal adversary. For the Boundary attack we used 1,000 iterations, a step-adaptation size of 1.5, and initial adversaries generated with the Blended Uniform Noise attack; for the Boundary attack we also evaluated the performance against one of the HBCNN (LeNet-5 variant) networks using 25,000, 100,000, and 1,000,000 iterations (tuning the step adaptation size for each), but this did not result in any notable improvement in its performance, and it remained uncompetitive against the HBCNN model in comparison to other attacks.

For Fashion-MNIST and CIFAR-10, surrogate models for creating transfer attacks were based on the respective standard and HBCNN architectures with label smoothing, but not noise, with the intent that these would be easier for the gradient-based methods to attack and generate adversaries from than the architecture augmented with both. Thus for these data-sets, each transfer attack was based on two surrogate models. For MNIST, the surro-

gate models for all attacks were based on : standard CNN with label smoothing, HBCNN with label smoothing, standard CNN+S+N, HBCNN+S+N, and CNN+AT models where two of each of these formed surrogates, with the exception of CNN+AT where only one was used, for a total of 9 surrogates. Note here that "+S+N" refers to the addition of label smoothing (S) and noise (N) in the training data. The two exceptions on MNIST to this were a) when targeting the vanilla CNN and HBCNN models, where vanilla CNN and HBCNN models replaced the standard CNN with label smoothing and HBCNN with label smoothing models and b) when targeting the ablated models for Figure 5.13, where any surrogates with the hierarchical binding architecture also had unpooling removed.

5.4.2 Geometrical Measures

In addition to the adversarial attacks we leverage, we perform two analyses of the geometric properties of our networks. These serve to both examine the relationship of our proposed architecture to the opening argument that adversarial robustness is dependent on an expanded neural manifold (Figure 5.1), as well as compare our approach to related work [Nassar et al., 2020]. These two analysis methods are described below.

5.4.2.1 Measuring Manifold Geometry

A manifold in neural state space represents the set of all points of activity that correspond to the same object, where the activity is determined in response to variations of the input, such as different object details, changes to scale, etc. [Chung et al., 2018]. Note the distinction to the data manifold, i.e. the manifold describing the concentration of example images in the embedding (e.g. pixel) space. To distinguish the two, we will refer to the first as the neural manifold. We have argued that the data manifold corresponding to all objects a human would classify as a given object is higher dimensional than often assumed, and that an architecture which can more faithfully represent the additional dimensions of variation (i.e. with a more extensive neural manifold) could support a more robust decision boundary. We are therefore interested in quantifying the geometrical properties of the neural manifolds in the networks we study. To do so, we apply the techniques developed in Chung et al. [2018] and Cohen et al. [2020]. This work used statistical mechanical theory to develop a novel measure of the *extent* of a neural manifold, a measure

influenced by both a manifold's dimension and radius.

In brief, these measures are developed from the structure of the hyperplane that separates manifolds in binary classification. Comparing any given manifold (e.g. the neural responses to cats) against another, there will be a particular point (the 'anchor point') on the cat manifold that will uniquely define a separating hyperplane to the other manifold. Multiple anchor points will exist for a given manifold to define its separation from other objects. For example, with a 2-dimensional, point-cloud manifold, one can imagine the points that an enclosing rubber band would touch as defining the convex hull of the manifold. The anchor points are those points in the convex hull that can define a linearly-separating boundary between that object manifold and the manifolds of other objects. Given a set of anchor points, statistical measures can then be applied to these points to estimate geometric properties. Of relevance to our work, the authors derived a measure of manifold radius (R_M), dimension (D_M), and extent ($R_M\sqrt{D_M}$). R_M captures the total variance of the anchor points, normalized by the average distance between manifold centres, while D_M captures the spread of the anchor points along the different manifold axes. Finally, $R_M\sqrt{D_M}$ combines these measures to characterize the manifold's total extent, and we use this quantity as our primary measure of manifold geometry.

To define these concretely, a manifold in neural state space can be defined as having $D + 1$ dimensions; one coordinate defines the centre of the manifold, while the other coordinates define the axes of variation. This manifold is embedded in the neural activity space of dimension N, where $D < N$, and the activity response to any given image is represented by the vector $\mathbf{x}$. The bold notation of $\mathbf{x}$ indicates that it is a vector in $\mathbb{R}^N$. The set $\mathcal{S}$ is the (in this case finite) set of data sample points that define the manifold. As any given point on the μ-th manifold M^μ lies in the lower, $D+1$-dimensional linear subspace, it can be parameterized as

$$\mathbf{x}^\mu(\overrightarrow{S}) = \sum_{i=1}^{D+1} S_i \mathbf{u}_i^\mu \tag{5.8}$$

Here, $\mathbf{u}_i^\mu$ are a set of orthonormal bases of the manifold's linear subspace, and like $\mathbf{x}$, these are N dimensional. The $D + 1$ scalars S_i then represent the coordinates of $\mathbf{x}^\mu$ within the

manifold's subspace, and the vector of coordinates is constrained to be in the set $\overrightarrow{S} \in \mathcal{S}$. Note that unlike the bold notation for $\mathbf{x}$ and $\mathbf{u}_i$, the arrow notation for $\overrightarrow{S}$ indicates that it is a vector in $\mathbb{R}^{D+1}$.

With these variables defined, recall that of interest is the separating hyperplane that enables classification of the objects, where there are a total of P objects/manifolds. In particular, the aim is to identify a hyperplane that provides the maximum separation capacity, given a separating margin of κ. The separating weight vector is then defined by up to P anchor points, that is, $w = \sum_{\mu=1}^{P} \lambda_\mu y^\mu \tilde{\mathbf{x}}^\mu$, where λ_μ is a multiplier satisfying $\lambda_\mu \geq 0$, y^μ is the binary label, and $\tilde{\mathbf{x}}^\mu \in conv(M^\mu)$, where $\tilde{\mathbf{x}}^\mu$ is the anchor point (denoted by tilde) in the convex hull of M^μ.

To characterize the geometry of the anchor points, they are first projected onto a lower $D+1$-dimensional subspace (where D is determined by the number of data samples per manifold, here 250). Using this projection of each anchor point $\tilde{\mathbf{x}}$ (denoted $\tilde{S}$), one can now define the measures of manifold geometry. In particular, the effective radius and dimension are defined in terms of $\tilde{\delta S} = (\tilde{S} - S_0)/\left\|S_0\right\|$. Here, $\tilde{\delta S}$ represents the projection of $\tilde{\mathbf{x}}$ onto the $D+1$-dimensional subspace, relative to the manifold's centre, S_0. The variation of these points is further normalized by the manifold centre norm. For each measure defined below, $\langle ... \rangle_{\overrightarrow{T}}$ represents an average over random vectors $\overrightarrow{T}$ of dimension $D+1$, and with i.i.d normally distributed components $T_i \sim \mathcal{N}(0,1)$. This statistical component is important, as the anchor points depend on the random orientations of the other object manifolds, and hence on $\overrightarrow{T}$. Bringing this together, the manifold radius is defined as the total variance of the normalized anchor points:

$$R_M^2 = \langle \left\| \tilde{\delta S}(\overrightarrow{T}) \right\|^2 \rangle_{\overrightarrow{T}} \tag{5.9}$$

The effective dimension, D_M, is defined as the angular spread between $\overrightarrow{\delta T} = \overrightarrow{T} - T_0$ (where T_0 is the projection of $\overrightarrow{T}$ onto S_0) and the anchor point $\tilde{\delta S}(T)$:

$$D_M = \langle (\overrightarrow{\delta T} \cdot \hat{\delta S}(\overrightarrow{T}))^2 \rangle_{\overrightarrow{T}} \tag{5.10}$$

Here $\hat{\delta S}$ is a unit vector in the direction of $\tilde{\delta S}$. Note that $D_M \leq D$. As noted earlier, the manifold extent is then given by $R_M\sqrt{D_M}$. Readers interested in additional mathematical details of these measures, including the determination of a given manifold's anchor points, are advised to refer to Chung et al. [2018] and Cohen et al. [2020].

We sample a total of 200 $\overrightarrow{T}$ in our analyses, as advised in the code-base of the implementation used - `https://github.com/schung039/neural_manifolds_replicaMFT` [Stephenson et al., 2019], and we use a margin value of $\kappa = 1$ for determining anchor points.

5.4.2.2 Eigenspectrum Analysis

We also apply the eigenspectrum analysis described in Stringer et al. [2019] and Nassar et al. [2020]. Specifically, given the covariance matrix $\sum$ of a network's layer, we analyse its eigenspectrum, denoted by the descending eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$. The covariance matrix is derived from activity sampled across multiple image presentations (here 750 examples per class in the MNIST test set). We then performed a linear fit to the eigenspectrum in log-log space, as described in Stringer et al. [2019], and implemented in the code-base for Nassar et al. [2020] available at `https://github.com/josuenassar/power_law`. This linear fit provides an estimate of the power-law exponent, α, which quantifies the decay of the eigenvalues (i.e. amount of described variance) as a function of the eigenvalue's rank. As we discuss further in our results, Stringer et al. [2019] observed that neural activity in mouse visual cortex approximately followed a power law such that the n-th eigenvalue (principal component) scaled as $1/n$, i.e. $\alpha = 1$.

As in Stringer et al. [2019], we limit the range of eigenvalues used for this fit; in our case, we limit the eigenvalues to 10 to 350, as our baseline architecture has a total of 400 eigenvalues, while the lower bound of 10 was used in Stringer et al. [2019], as it was above this eigenvalue that they observed the power-law.

5.5 Results : Hierarchical Binding Causes Multiple Adversarial Attacks to Fail

In the following section, we begin by demonstrating that simply introducing hierarchical binding is insufficient for an improved decision boundary, but that additional steps of regularization and noise in the training data provide a more robust model. After describing the settings in which we observe improved robustness, we use additional analyses to clarify the role that the expanded manifold due to hierarchical binding plays in this observed effect.

5.5.1 Hierarchical Binding Alone

Our opening assumption is that the true manifold representing objects is higher dimensional than typically assumed when designing CNNs, and that capturing this would improve robustness. In particular, the HBCNN architecture affords the possibility of learning better decision boundaries along additional dimensions of variation. Importantly however, the data in a typical data-set such as MNIST will often be concentrated on a lower-dimensional manifold than is likely to underlie human perception. Furthermore, decision boundaries in high dimensions are challenging to learn due to the sampling complexity (i.e. the chance of sampling along a particular dimension becomes vanishingly small, and so the amount of information available to learn the decision boundary in these dimensions is limited). We begin by discussing our results in the context of the HBCNN (LeNet-5 variant) leveraged on the MNIST data-set.

Given the above, what then is the effect of introducing binding representations without any additional modifications? Given the sparse sampling in the high-dimensional space, we would expect the augmented model's decision boundary to have limited, if any improvement (Figure 5.1b), as it attempts to fit to the few data-points available. These decision boundaries however have a chance of covering directions that are otherwise vulnerable to transfer attacks, and consistent with this, we observe some enhanced robustness to L_2 transfer attacks (Table 5.2). This is notable, as the inclusion of hierarchical binding alone appears to cause minimal, if any, gradient masking (demonstrated by the effectiveness

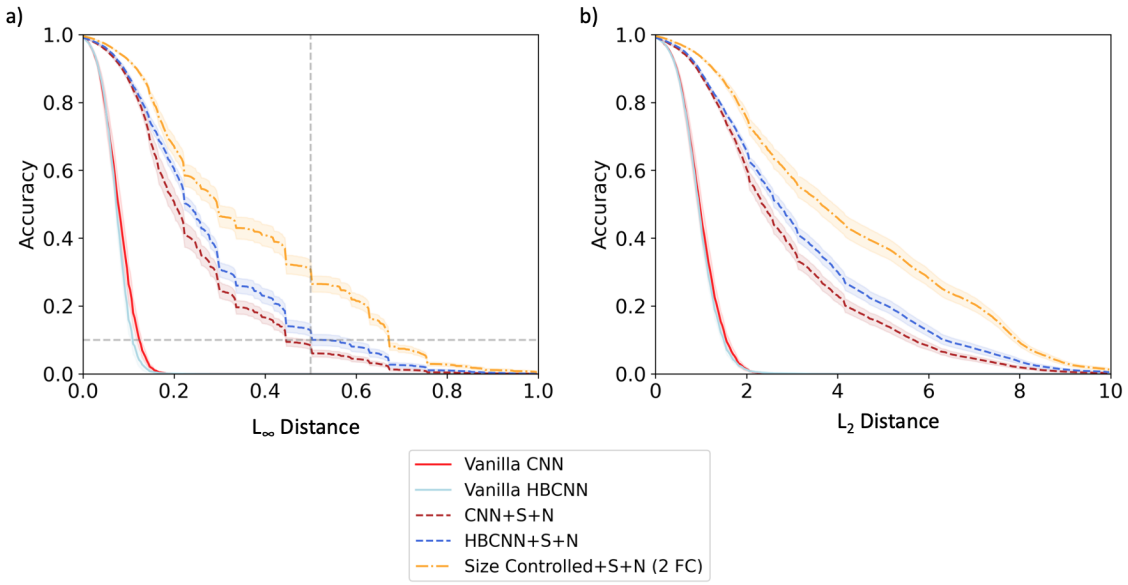


Figure 5.4: *Distortion-accuracy curves for the vanilla and augmented (label smoothing and noisy training data) models.* The y-axis shows the accuracy of the model as the allowable distortion for the attack increases. a) L_∞ PGD attack, b) L_2 PGD attack. Accuracy above chance (horizontal grey line) on the L_∞ PGD attack beyond a distortion of 0.5 (vertical grey line) provides evidence of gradient masking. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training. Shaded areas indicate 95% confidence interval of the mean across the 30 sample networks for each architecture.

of the gradient-based PGD attack in Figure 5.4). As such, there is nothing to prevent the creation of effective transfer attacks from our surrogates [Tramèr et al., 2020b]. This suggests that across the variety of directions that an image can be perturbed to fool a classifier (including our own model), these are on average *more* effective against an undefended model.

In addition to the enhanced L_2 transfer attack resistance, some enhanced robustness to white-box attacks that use estimated gradients is seen (Table 5.2), likely as a result of a less smooth loss-landscape caused by the gradient unpooling and unpooling operations, similar to the effect in Xiao et al. [2020]. Unlike in Xiao et al. [2020] however, the effects are minimal and sufficiently small so as to not impact white-box attacks with direct access to gradients. The thresholding that our gradient unpooling and unpooling operations introduce also increases resistance to the Boundary Attack, although this is only due to a failure of its ability to operate in its normal regime, rather than an indication of genuinely enhanced robustness. In particular, as the Boundary Attack attempts to navigate the deci-

sion boundary, it frequently becomes stuck in local minima and encounters regions where it must decrease its step size for the boundary to behave linearly.

The above results demonstrate an improvement on L_2 transfer attacks, and that elements of the architecture can inadvertently compromise the optimization techniques of some attacks. However, it was also predicted that, in the absence of additional samples, the model would be unable to learn a particularly useful decision boundary. Consistent with this, the trend is that hierarchical binding alone does not enhance robustness across the majority of attacks, and in fact some attacks are more successful against the LeNet-5 variant of the HBCNN than against an unaltered LeNet-5 model (Table 5.2); clearly the decision boundaries are not yet very useful and can in fact be worse than those of a standard model.

5.5.2 Hierarchical Binding with Regularization and Noisy Training Data

To better sample the high-dimensional data manifold corresponding to human perception, we introduce Gaussian ($\sigma = 0.3$) and salt-and-pepper (120/784 pixels perturbed) noise during training (Figure 5.1c). This is motivated by previous work where training with noise has been associated with increased robustness to adversarial attacks, measured either by the average adversary distance or the accuracy at a fixed-threshold perturbation [Zantedeschi et al., 2017, Ford et al., 2019, Rusak et al., 2020]. Alongside the introduction of noise, we include regularization in the form of label smoothing, where it is expected that this will result in a smoother decision boundary that better generalizes to novel data-points. Finally, for both the control model and the HBCNN, we also increase the dimension of the fully connected layers to 256 and 128, and train for longer, as cross-validation data suggested this better modelled the more complex data-set introduced by the addition of noise, and further enhanced robustness. Owing to these changes, comparisons between the vanilla and augmented models are of limited value, but as we will highlight, our primary interest is in comparing the HBCNN model to the control models.

As shown in Table 5.3, this creates a network ('HBCNN+S+N' in the table) with enhanced robustness to virtually all attacks relative to the main control model. This result supports

Table 5.2: MNIST Results (Part 1)

	Vanilla-CNN	Vanilla-HBCNN
Clean	99.25	99.13
L_2-metric ($\epsilon = 1.5$)		
Transfer ■	2.4 (84%)	2.6 (86%)
Uniform Noise ■	9.0 (99%)	8.7 (99%)
Gaussian Noise ■	7.1 (98%)	5.9 (98%)
Boundary ■	1.6 (57%)	7.2 (98%)
Pointwise ■	3.4 (96%)	2.9 (93%)
FGM	3.0 (82%)	2.8 (78%)
FGM w/GE	3.3 (82%)	6.6 (91%)
DeepFool	1.4 (46%)	1.4 (39%)
DeepFool w/GE	1.7 (59%)	1.6 (54%)
BIM	1.3 (37%)	1.3 (34%)
BIM w/GE	1.2 (34%)	1.5 (51%)
PGD	1.0 (13%)	1.0 (12%)
All L_2	1.0 (12%)	1.0 (12%)
L_∞-metric ($\epsilon = 0.3$)		
Transfer ■	0.21 (13%)	0.21 (9%)
FGSM	0.18 (15%)	0.18 (20%)
FGSM w/GE	0.23 (38%)	0.40 (54%)
DeepFool	0.12 (0%)	0.12 (0%)
DeepFool w/GE	0.14 (0%)	0.13 (1%)
BIM	0.10 (0%)	0.10 (0%)
BIM w/GE	0.10 (9%)	0.11 (29%)
MIM	0.10 (0%)	0.10 (0%)
MIM w/GE	0.10 (9%)	0.13 (27%)
PGD	0.08 (0%)	0.08 (0%)
All L_∞	0.08 (0%)	0.08 (0%)
L_∞-metric ($\epsilon = 12$)		
Pointwise $\times 10$ ■	9 (28%)	7 (14%)
Salt&Pepper Noise ■	56 (93%)	22 (73%)
All L_0	9 (28%)	7 (14%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the MNIST data-set; note that the main MNIST results are split across two tables (this table and Table 5.3) due to page-width constraints. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing or noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); w/ GE = with Gradient Estimation; ■ = black-box attack.

the notion that introducing hierarchical binding can support developing a more robust model. As we do not perform adversarial training in our models, we cannot guarantee that any particular model will have developed a robust decision boundary. As such, we observe that our models follow a distribution in robustness, and in addition to the tabular results, we present distortion curves (Figure 5.5) and histograms (Figure 5.6) of model robustness for several key attacks. Figure 5.5 in particular reveals that the primary robustness benefit of the HBCNN+S+N emerges at higher values of distortion. We include example images that the HBCNN+S+N has misclassified at varying amounts of noise to demonstrate that the improved robustness is not associated with (implausible) super-human performance.

The robustness we observe to such a wide range of black and white-box attacks suggests that gradient masking alone cannot explain our results. Furthermore, our distortion-accuracy plot in Figure 5.4 demonstrates that with an L_∞ perturbation budget of 0.5 (sufficient to change every pixel-value to grey), the accuracy of the HBCNN models reassuringly reach that of random guessing [Carlini et al., 2019]. Despite this, some of our white-box attack results suggest that in both the control ('LeNet+S+N' in Table 5.3) and HBCNN+S+N, a degree of gradient-masking is present. This appears to be an unintended consequence of label smoothing, as well as the discontinuities introduced by the thresholding operation in unpooling and gradient-unpooling [Xiao et al., 2020].

We emphasize that the interesting result is the broad range of enhanced robustness seen, in particular to black-box attacks, and across multiple L_p norms. With the exception of the Boundary Attack, these black-box attacks rely on sampling directions from the input to find an adversarial region, and the improved robustness points to the effectiveness of hierarchical binding in reducing the vulnerability of these directions.

We note that the transfer attack robustness, while small in magnitude, is particularly significant given that we leverage such a broad transfer attack against the MNIST-trained models, and given the relationship of transfer attacks to our opening geometric intuition. Many transfer attack evaluations in the literature use a single model, or even one with a different architecture to generate the transfer images. We use a total of 9 surrogate models and 18 adversarial candidates for each attack, including surrogates that have been adversarially trained and independent equivalents of the architecture being attacked. Using so

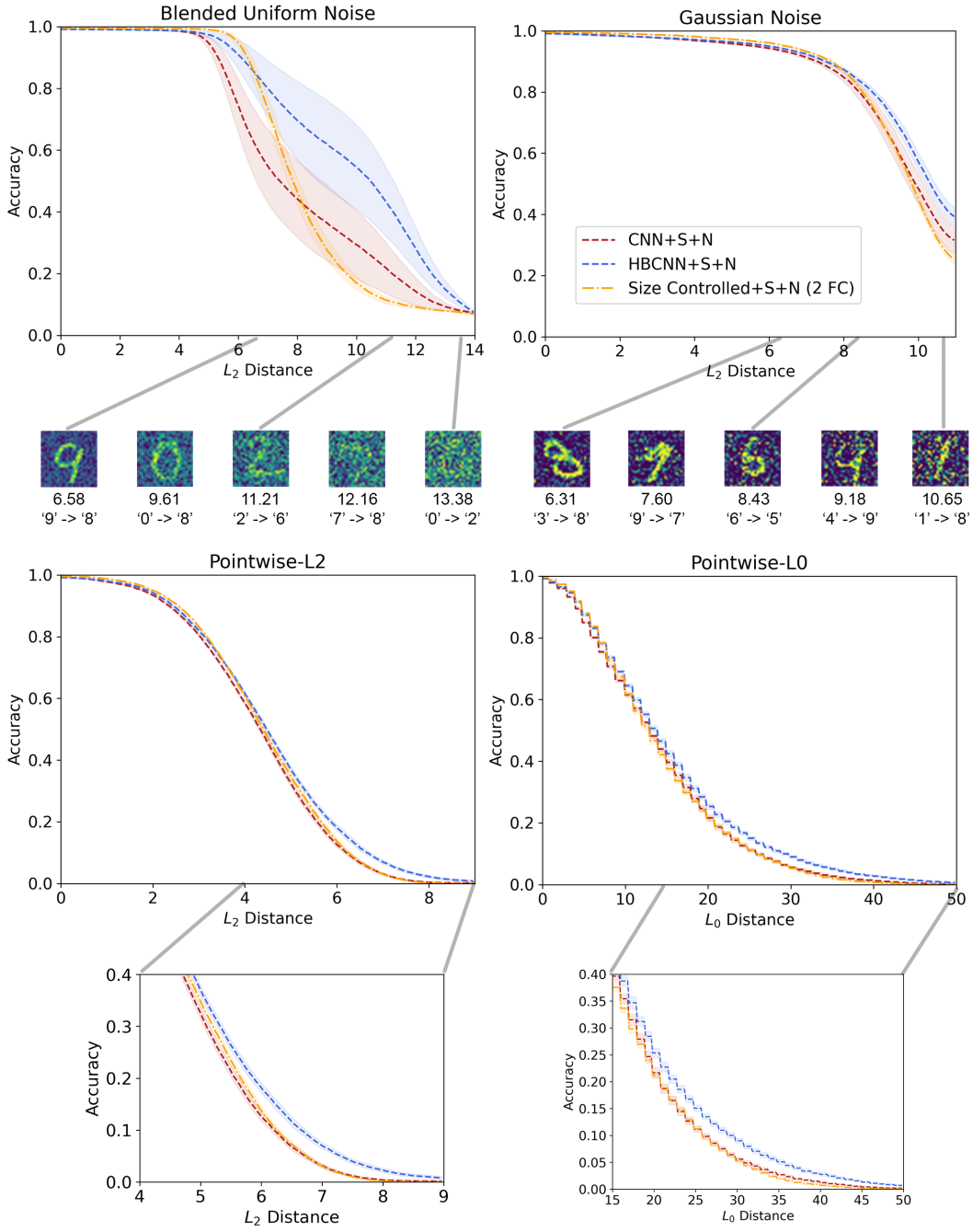


Figure 5.5: *Distortion-accuracy curves for the augmented (label smoothing and noisy training data) models on black box attacks.* The y-axis shows the accuracy of the model as the allowable distortion for the attack increases. The digits displayed under Blended Uniform and Gaussian noise indicate example adversaries that successfully fooled the HBCNN+S+N model, providing both their L_2 distance, as well as the change in classification. For the Gaussian Noise attack, the attempted additive noise does not extend beyond an L_2 distance of 11. The plots expanded from the Pointwise attacks show zoomed-in views of the distortion curves. The shaded regions indicate the 95% confidence interval of the mean across the 30 sample networks for each architecture.

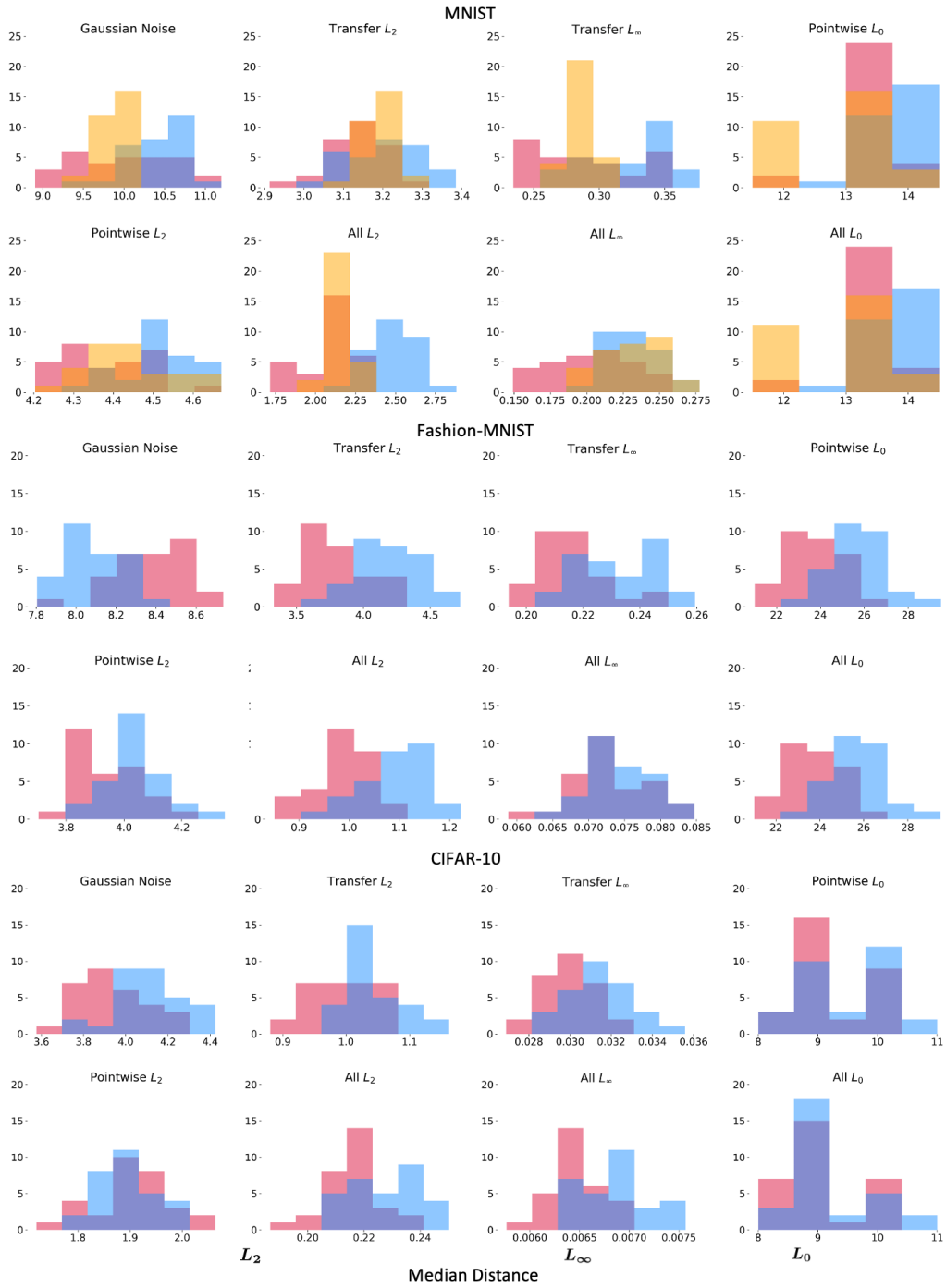


Figure 5.6: *Distributions of the Median Performance of Networks*. Shown are the distributions of the median performance of networks (30 for each condition). Blue indicates results for the HBCNN (LeNet-5 or VGG variant as appropriate for the data-set) + label smoothing (S) + noise in the training data (N); red represents the LeNet-5/VGG+S+N as appropriate for the data-set; yellow indicates the Size-controlled CNN+S+N (2 fully connected layers). The x-axis for each distribution indicates the distance between the original image and the adversary, using the distance measure appropriate for that attack. Note that we do not show results for every attack leveraged, but select those we deemed particularly important such as black-box attacks and the 'All' attacks evaluations.

many candidates in our evaluation is important, as our defence is proposed to reduce the number of directions in which our HBCNN architecture and other models share vulnerable decision boundaries, which transfer attacks are otherwise designed to exploit [Tramèr et al., 2017]. The observed improvement in robustness therefore lends evidence to our opening claim that, while a sufficiently powerful attacker can break our defence, these directions are more challenging to find in the HBCNN than in a standard model.

5.5.3 Comparisons to Additional Models on MNIST

To control for the possibility that the greater number of parameters in the HBCNN simply enables it to fit the noisy training data better, we also train two larger CNNs with an equivalent number of parameters to the HBCNN (results in Table 5.3). The first version has a single fully connected layer, as is common in adversarially robust benchmark CNNs, such as the adversarially trained model in Madry et al. [2018]. The second has two fully connected layers, like the baseline CNN and the HBCNN model. The performance of both architectures is in general not comparable, and in some cases, the larger models are more vulnerable than the smaller, baseline architecture. For the stronger, size-controlled CNN with two fully connected layers, this is the case for several important black-box attacks we evaluate, including the transfer L_∞, Boundary, and additive Gaussian noise attacks. These results indicate that arbitrarily adding parameters can increase the vulnerability of a model.

The primary exceptions to the enhanced vulnerability of the size-controlled (2 fully connected layers) model are the results of several gradient-based attacks, and the transfer L_2 attack. Regarding gradient-based attacks, this difference appears to be a simple consequence of a significant increase in gradient masking in the size-controlled model, an uninteresting defence. Evidence of gradient masking in this model includes that a distortion budget of $L_\infty = 0.5$ is insufficient to bring accuracy to chance levels (Figure 5.4). This is significant, as the convergence plots of the PGD attacks in Figure 5.7 demonstrate that the PGD attack is as well optimized as possible, and such a distortion budget should enable the attacker to change every pixel to a grey-scale value (i.e. making above-chance accuracy impossible unless there is gradient masking). The existence of considerable

gradient masking is further supported by the observation that the PGD attacks perform worse than the equivalent transfer attacks for this model, and the significantly better performance of the Boundary Attack in comparison to PGD (Table 5.3). For the transfer L_2 result, we highlight that the size-controlled models were not used as surrogate models when generating attacks for transfer (owing, in part, to the significant gradient masking), and so it is promising that the HBCNN still at least matches the robustness of the deeper size-controlled model under this setting. More importantly, in spite of this limited transfer attack against the size-controlled model in comparison to the HBCNN model, the HBCNN model is considerably more robust on transfer L_∞ attacks. As such, any apparent robustness in the size-controlled (2 fully connected layers) model appears to be a result of gradient masking, which as we have noted, is an uninteresting defence. In terms of why this model displays a large degree of gradient masking, this is likely again an unintended consequence of label smoothing, and its complex interactions with the properties of any specific model. In summary, these results support the inductive bias in our architecture that abstract and low-level features should be represented in their own right; it is not a question of simply adding more free parameters to attain robustness.

We next compare our results to models that have undergone adversarial training. Adversarial training (AT) is a process of optimizing the parameters of a network given both clean training examples and adversarial images generated for that network. This process continues in an iterative fashion, with new adversaries generated as the model attempts to develop a robust decision boundary, and this remains one of the strongest methods for defending classifiers [Madry et al., 2018, Gowal et al., 2020].

We highlight that our 'HBCNN+S+N' model beats adversarial training on several black-box attacks, the All L_2 metric, and the All L_0 metric. These results are consistent with the observation that adversarial training can reduce robustness to noise [Dapello et al., 2020, Rusak et al., 2020, Geirhos et al., 2021], and black-box attacks more generally [Schott et al., 2019]. We also emphasize that the adversarially trained model from Madry et al. [2018] uses many more parameters; whereas our method has around 850,000 parameters, the AT model is nearly four times as large with approximately 3,275,000 trainable parameters. Despite our method achieving comparable or better robustness on several metrics

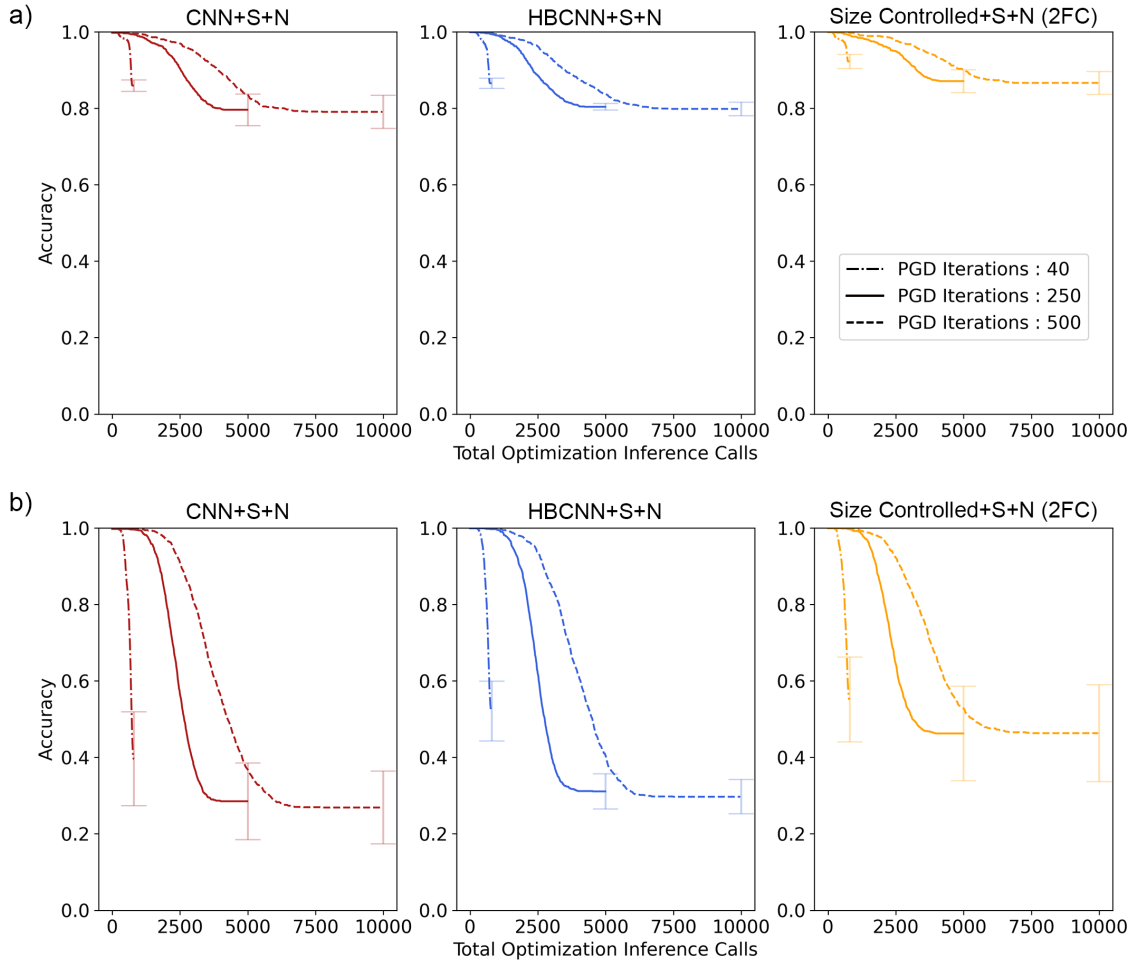


Figure 5.7: *Convergence of PGD attacks nested in binary search over hyperparameters.* (a) and (b) demonstrate the convergence of our PGD attacks for the key architectures we compare on MNIST. Demonstrating convergence is required in order to attribute improbable robustness to white-box attacks to the phenomenon of gradient masking, as in the case of the Size-Controlled+S+N (2FC) model. The y-axis shows the accuracy of the model as the total number of inference calls increases, where accuracy is provided at a threshold of $\epsilon = 1.5$ for the L_2 PGD attack (a), and $\epsilon = 0.3$ for the L_∞ PGD attack (b). Each PGD attack is associated with 20 random starts, of which the best result for each image is used. For each random start, a binary search with $k = 20$ steps is performed to optimize the step-size and ϵ value of the PGD attack, minimizing the final perturbation of a successful adversary. Each binary search iteration is associated with a budget of PGD iterations (n = 40, 250, or 500). Thus, excluding the initial model queries to establish a starting point for the binary search, there are a total of $n \times k$ possible inference calls for a given optimization process. We show the convergence of the optimization process given different PGD budgets, where each setting has the flexibility to tune the hyperparameters of the attack. While increasing the possible number of PGD iterations from a typical value of 40 to 250 provides a marked benefit, doubling this to 500 does not. We therefore choose a value of 250, which also enables us to run the PGD attack across many seeds without a prohibitive computational cost. Shaded regions represent the 95% confidence interval across 5 random seeds.

using significantly fewer parameters, we also observe higher classification accuracy on the clean data-set (99.05% for our model vs 98.40% for the AT model).

We do not include the robust Analysis by Synthesis (ABS) model [Schott et al., 2019] due to the computational resources required to run it (around three orders of magnitude more time for a forward pass on our GPUs in comparison to the HBCNN). Comparing the results in Table 1 of Schott et al. [2019], with those of the 'HBCNN+S+N' model in our Table 5.3, our network outperforms their non-binary ABS model's state-of-the-art (SOTA) All L_2 result (median distance 2.3 for ABS vs median distance 2.5 for ours), and achieves similar results on Gaussian noise and Transfer L_∞ attacks. Our results are thus in keeping with SOTA robustness on MNIST according to the All L_2 evaluation, however we do not make any such claim due to the difficulties of a fair, head-to-head comparison, and limitations of our defence that we raise in the discussion.

In spite of the above promising results, it is clear that the HBCNN+S+N is still far from achieving the robustness of a human to adversarial examples. On occasion, semantically meaningful adversarial examples do emerge, but these do not appear, on average, to be more prevalent for the HBCNN+S+N than those for the control models. For example, from amongst 300 MNIST adversarial examples for HBCNN+S+N and CNN+S+N models, we selected what we thought to be (in our potentially biased view) the 10 most semantically convincing adversaries generated by the BIM L_2 attack for each model (Figure 5.8). We feel there are convincing examples for both models, with no obvious systematic difference.

5.6 Results : Connecting the Geometry of Representations to Hierarchical Binding and Robustness

5.6.1 Manifold Extent and Robustness

We have suggested that vulnerability to adversarial examples can be reduced by developing models that more faithfully represent the high-dimensional manifolds that underlie human perception. We therefore proposed an architecture with the intent of better mod-

Table 5.3: MNIST Results (Part 2)

	CNN+S+N	HBCNN +S+N	Sized-1FC +S+N	Sized-2FC +S+N	CNN+AT
Clean accuracy	99.10%	99.05%	99.43	**99.45%**	98.40%
L_2-metric ($\epsilon = 1.5$)					
Transfer ■	3.1 (89%)	**3.2** (89%)	3.0 (92%)	**3.2** (92%)	3.7 (95%)
Uniform Noise ■	6.6 (99%)	**11.3** (99%)	6.2 (99%)	7.9 (100%)	8.7 (98%)
Gaussian Noise ■	10.0 (99%)	**10.5** (99%)	9.7 (99%)	9.9 (99%)	5.3 (97%)
Boundary ■	2.5 (88%)	**9.2** (98%)	2.1 (84%)	2.3 (88%)	1.4 (42%)
Pointwise ■	4.4 (96%)	**4.5** (97%)	4.2 (97%)	4.4 (97%)	1.9 (73%)
FGM	8.4 (92%)	9.7 (93%)	**9.8** (94%)	8.6 (96%)	∞ (95%)
FGM w/GE	8.2 (92%)	∞ (95%)	9.6 (94%)	8.3 (96%)	∞ (95%)
DeepFool	7.3 (91%)	8.1 (93%)	3.6 (89%)	**9.3** (96%)	9.4 (94%)
DeepFool w/GE	7.3 (93%)	7.7 (94%)	4.6 (91%)	**9.7** (96%)	9.5 (94%)
BIM	3.6 (84%)	4.0 (86%)	3.0 (85%)	**4.6** (92%)	4.9 (93%)
BIM w/GE	3.6 (84%)	4.0 (87%)	3.1 (84%)	**4.5** (92%)	4.5 (93%)
PGD	2.5 (77%)	2.8 (79%)	1.9 (71%)	**3.7** (87%)	2.8 (86%)
All L_2	2.1 (75%)	**2.5** (78%)	1.8 (69%)	2.1 (81%)	1.4 (39%)
L_∞-metric ($\epsilon = 0.3$)					
Transfer ■	0.29 (43%)	**0.33** (60%)	0.26 (33%)	0.28 (41%)	0.39 (94%)
FGSM	0.48 (76%)	**0.62** (82%)	0.50 (76%)	0.46 (73%)	0.44 (95%)
FGSM w/GE	0.50 (78%)	**0.63** (83%)	0.51 (76%)	0.47 (79%)	∞ (95%)
DeepFool	0.83 (81%)	**1.0** (85%)	0.32 (54%)	**1.0** (94%)	0.46 (94%)
DeepFool w/GE	**1.0** (86%)	**1.0** (86%)	0.45 (73%)	**1.0** (95%)	0.71 (94%)
BIM	0.34 (56%)	0.44 (66%)	0.25 (31%)	**0.45** (63%)	0.36 (93%)
BIM w/GE	0.34 (57%)	**0.45** (67%)	0.25 (31%)	**0.45** (62%)	0.63 (93%)
MIM	0.32 (54%)	**0.42** (66%)	0.25 (33%)	0.41 (59%)	0.34 (93%)
MIM w/GE	0.35 (56%)	**0.44** (68%)	0.26 (38%)	0.42 (60%)	0.44 (94%)
PGD	0.22 (28%)	0.24 (33%)	0.15 (0%)	**0.30** (49%)	0.33 (91%)
All L_∞	0.20 (20%)	0.22 (25%)	0.15 (0%)	**0.23** (25%)	0.33 (91%)
L_0-metric ($\epsilon = 12$)					
Pointwise $\times 10$ ■	13 (53%)	**14** (56%)	12 (46%)	13 (52%)	4 (0%)
Salt&Pepper Noise ■	142 (97%)	135 (97%)	**156** (98%)	150 (98%)	14 (57%)
All L_0	13 (53%)	**14** (56%)	12 (46%)	13 (52%)	4 (0%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the MNIST data-set; note that the main MNIST results are split across two tables (this table and Table 5.2) due to page-width constraints. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with label smoothing and noise added to the training images; blue indicates the best performance across all networks. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); Sized-nFC = size-controlled CNN with either $n = 1$ or 2 fully-connected layers; AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/ GE = with Gradient Estimation; ■ = black-box attack.

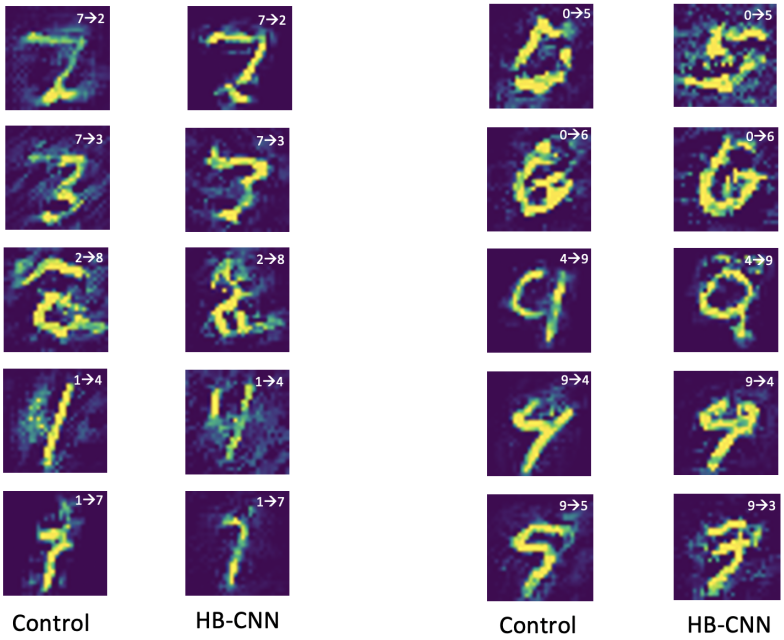


Figure 5.8: *Cherry-picked, semantically meaningful adversaries.* The annotation shows the original class followed by the prediction of the network following the adversarial perturbation. Shown are adversaries for the control (LeNet-5) and HBCNN (LeNet-5 variant) networks with label smoothing and noisy training data.

elling additional degrees of low-level variation. We subsequently trained with noise, in the hope of sampling more densely from the data manifold that corresponds to human representations of hand-written digits, as opposed to the more constrained data manifold found in the original MNIST training set. While these modifications were associated with enhanced robustness, it is important to determine whether our modified architecture does indeed model greater degrees of object variation, and the degree to which this is associated with robustness.

To measure the geometry of the neural manifolds in our different architectures, we apply the techniques of Chung et al. [2018] and Cohen et al. [2020]. In particular, these enable us to evaluate the average dimension (D_M), radius (R_M), and extent ($R_M\sqrt{D_M}$) of the neural manifolds for the object classes. Based on our opening proposal, we would expect the neural manifolds in the hierarchical binding network's layer immediately proceeding the fully connected layers of the network to be higher dimensional than in the other architectures. Furthermore, we would expect the HBCNN to be better able to model the natural variation of objects across these dimensions, which would manifest as a larger av-

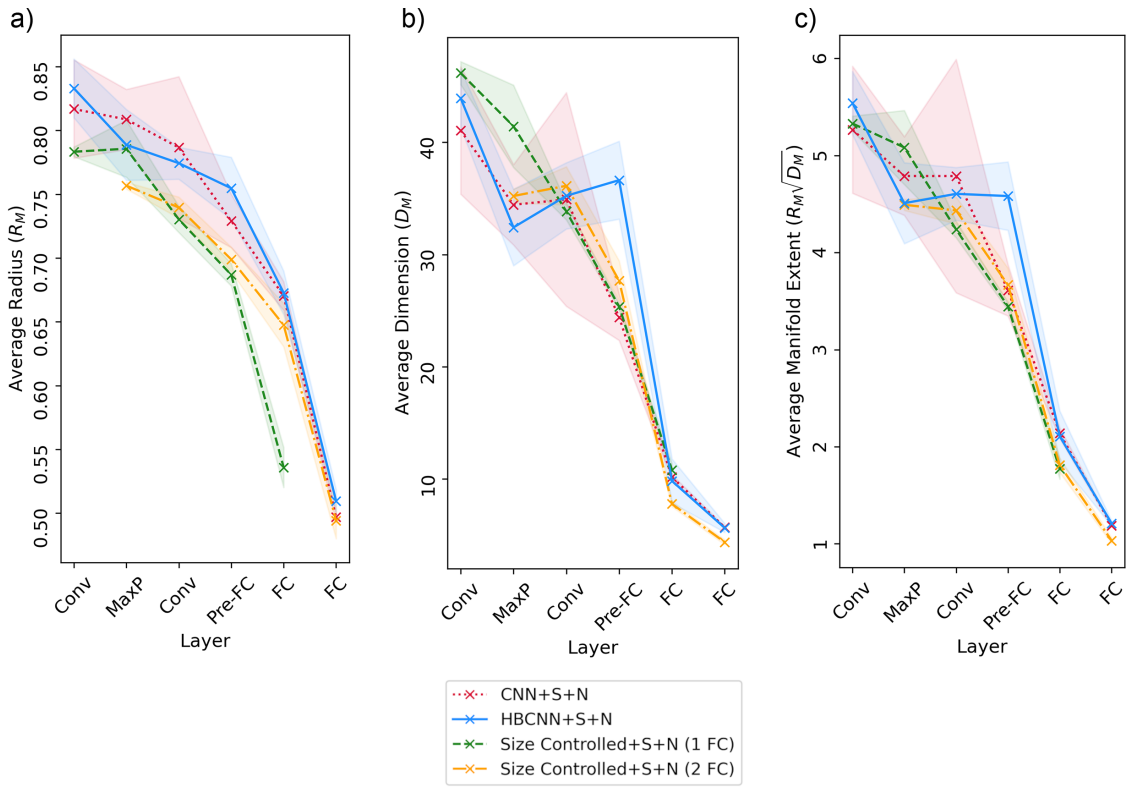


Figure 5.9: *Manifold Properties as a Function of Layer.* Presented is the average neural manifold a) radius, b) dimension, and c) extent as a function of layer for the architectures we evaluate on MNIST. Note that the Size Controlled+S+N (2 FC) model has no result for its first layer, owing to insufficient GPU memory for determining this. Values represent the mean across 5 sample networks, while shaded regions indicate the 95% confidence interval. Conv = post convolution and ReLU operation; MaxP = post max-pool operation; FC = post fully-connected weights and ReLU operation. For the non-binding models, the Pre-FC layer is equivalent to MaxP. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

erage manifold radius. The manifold extent captures (and is proportional to) both of these metrics, and should therefore also be increased under our opening proposal. Consistent with this, we observe that all of these measures are elevated in the HBCNN relative to the other architectures in the layer immediately proceeding the fully connected parts of the networks (Figure 5.9).

Furthermore, our proposal suggests that expanded neural manifolds would be associated with robustness to the measures we are most interested in, namely robustness to black-box attacks. We therefore estimate the manifold extent in the layer of activity immediately proceeding the fully connected layers, and compare this to our measures of robustness.

As predicted, across all of these attacks, and importantly *across all three metrics of L_2, L_∞, and L_0 adversaries*, we observe a positive relationship between manifold extent and robustness (Figure 5.10). This provides evidence that, not only does the introduction of hierarchical binding expand the neural manifold, but that this modification is associated with the improvement in robustness observed in the HBCNN.

It is also worth highlighting that, although expanding the manifold is associated with a more robust decision boundary, it does not actually guarantee the robustness of said boundary. For example, it is clear from our results that even models with small manifold extents can occasionally be robust. Our results therefore do not support that expanded manifolds are necessary, nor sufficient, for robustness in the setting that we explore. Importantly however, there is a clear association between models with expanded manifolds and robustness.

It is also interesting to note that, comparing the two size-controlled CNN architectures, the network with two-fully connected layers appears less constrained to rapidly minimize its manifold radius, dimension, and extent across the layers (Figure 5.9), and this might partly explain some of its greater robustness. This is noteworthy, as many CNNs in the adversarial examples literature, such as the robust model in Madry et al. [2018], or the baseline control models in Nassar et al. [2020] and Schott et al. [2019], use a single fully connected layer.

As an additional point of interest, it is worth examining where the adversarially trained model from Madry et al. [2018] sits in these evaluations. With only one sample, and robustness measures that are significant outliers, we mark the location of the model on our scatter plots with a vertical grey line (Figure 5.10). We note that the manifold extent is above virtually all of the non-HBCNN models, a finding that is consistent with the positive relationship we observe between manifold extent and robustness.

5.6.2 Power Law Scaling of Variance

Following on from the above analysis, it is interesting to compare our proposal to another possible connection between the nature of neural manifolds and robustness. In Stringer et al. [2019], it was observed that the neural responses in mouse visual cortex were high-

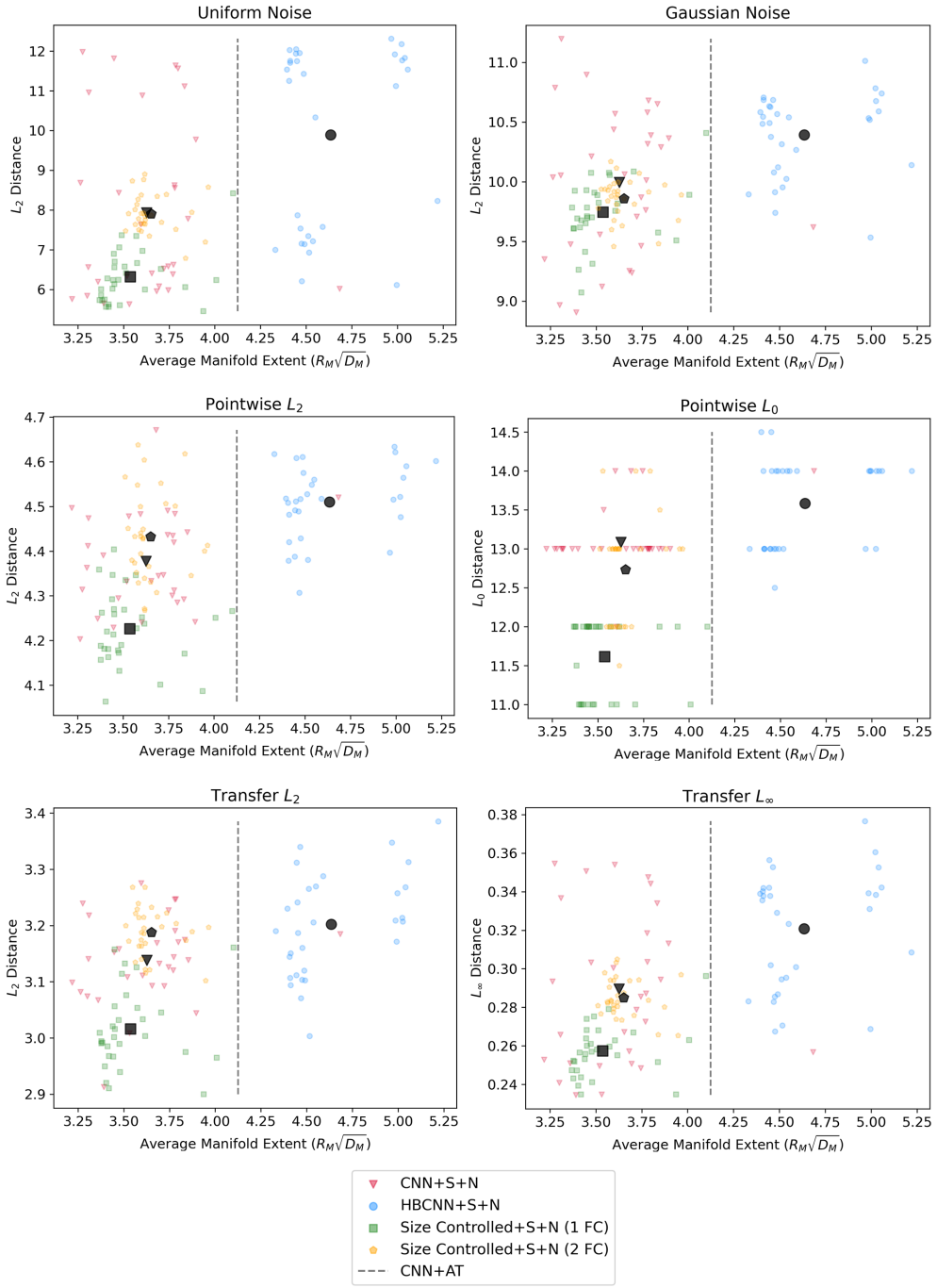


Figure 5.10: *Robustness to Black-Box Attacks vs. Manifold Extent.* The robustness of individual models as a function of their average manifold extent ($R_M\sqrt{D_M}$). Manifold extent is determined for the layer of activity immediately proceeding the fully connected layers. The black markers indicate the mean value for a given architecture type. The dashed grey-line indicates the $R_M\sqrt{D_M}$ value of the adversarially trained model from Madry et al. [2018], which is not displayed as a point due to its significant outlier values for robustness. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

dimensional, and that the variance of the n-th dimension scaled according to the power law $n^{-\alpha}$, where $\alpha \approx 1$. Stringer et al. [2019] proposed that the brain maintains a high-dimensional representation of the input, while ensuring that it decays sufficiently quickly that the underlying manifold is smooth. If the representation decayed more slowly, then the manifold would be non-differentiable, which could manifest in sudden changes in responses given small shifts in the input space. For this reason, they proposed that a power law decay with $\alpha \approx 1$ could support robustness, including to adversarial examples. In Nassar et al. [2020], this proposal was directly tested in deep neural networks by implementing a form of regularization that explicitly promoted the eigenspectra of neural responses to satisfy $\alpha = 1$. They observed that the introduction of such regularization promoted robustness to both white-box attacks and general noise corruption, suggesting that this could partially explain the robustness of biological vision.

It is reasonable to wonder whether the introduction of hierarchical binding results in a similar change as the regularization term used in Nassar et al. [2020]. If it does, this would be a welcome outcome, as the regularization term in Nassar et al. [2020] is computationally intensive to implement, which limited their approach to MNIST. Furthermore, it is not clear how this power law spectrum would manifest naturally in the brain (i.e. without artificially punishing representations with a regularization term).

We therefore used the same approach applied in Stringer et al. [2019] to fit an α exponent to the eigenspectrum of each of our networks, using the layer of activity immediately proceeding the fully connected layers. Some example eigenspectra are provided in Figure 5.11. Interestingly, the α exponent is indeed closer to 1 in the HBCNN (mean 1.27 ± 0.02, 95% CI) vs. the baseline LeNet model (mean 1.57 ± 0.01, 95% CI), or (included for comparison), the base-line CNN architecture used in Nassar et al. [2020] (1.46, result from a single model). However, this is not unique to the HBCNN, and in general, the α exponent appears, if anything, closer for our two size-controlled models, the size-controlled CNN with a single fully-connected layer (mean 1.24 ± 0.02, 95% CI), and with two fully connected layers (mean 1.245 ± 0.005, 95% CI). Consistent with this, we do not observe an obvious relationship across the architecture types between the α exponent and any of our primary black-box robustness metrics (Figure 5.12). While some of the scatter plots, such

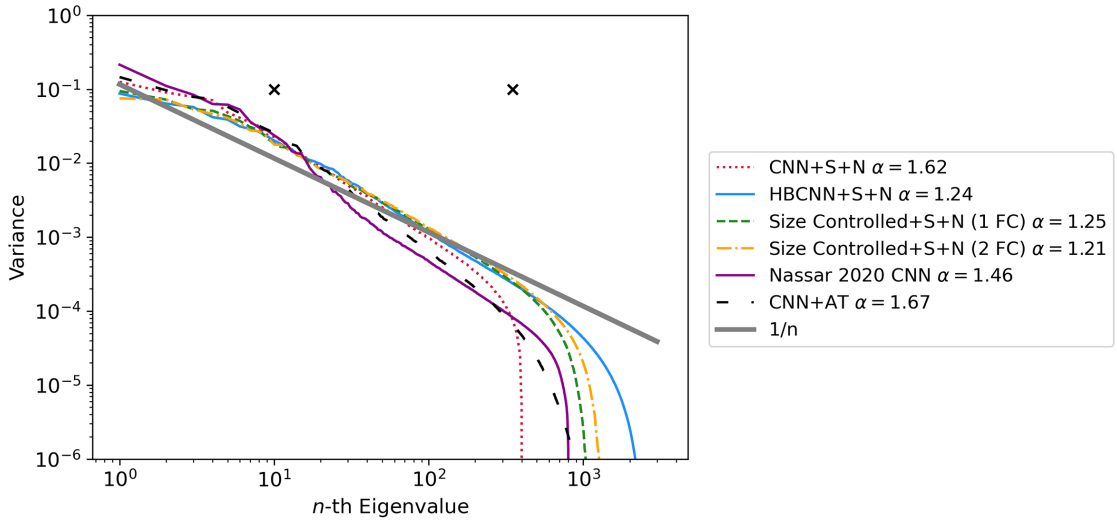


Figure 5.11: *Sample Eigenspectra for Median Models Trained on MNIST.* The eigenspectra of sample models, where the model with the median (or nearest to median) α value for that architecture is selected for plotting. Also shown are the eigenspectra for the adversarially trained model [Madry et al., 2018], and a CNN with the same architecture and hyperparameters as the baseline model in Nassar et al. [2020], for comparison. The '$\times$' symbols indicate the range of ranked eigenvalues used for estimating the α exponent (10, 350), where the total number of eigenvalues in the CNN+S+N model is 400. The plotted eigenvalues are normalized by the total variance. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

as for the Gaussian noise attack, suggest possible correlations within a given architecture, these do not appear to be in consistent directions. In summary, and unlike the measure of manifold extent related to our opening proposal, there is no evidence that a power-law spectrum might explain the robustness changes seen across our architectural variants.

We noted earlier that the adversarially trained model in general has an elevated manifold extent compared to typical networks. It is interesting to therefore wonder where the estimated α exponent of the adversarially trained model from Madry et al. [2018] sits. Curiously, we find that the model actually has a *larger* α exponent than any of the other models (Figure 5.12), where we once again mark the adversarially trained model with a grey line. This may seem counter-intuitive given the result in Nassar et al. [2020] suggesting that standard architectures have larger α exponents, and that regularizing them to fall closer to 1 improves robustness. It is, however, consistent with the original proposal in Stringer et al. [2019] that a larger α exponent would result in a smoother (and therefore a

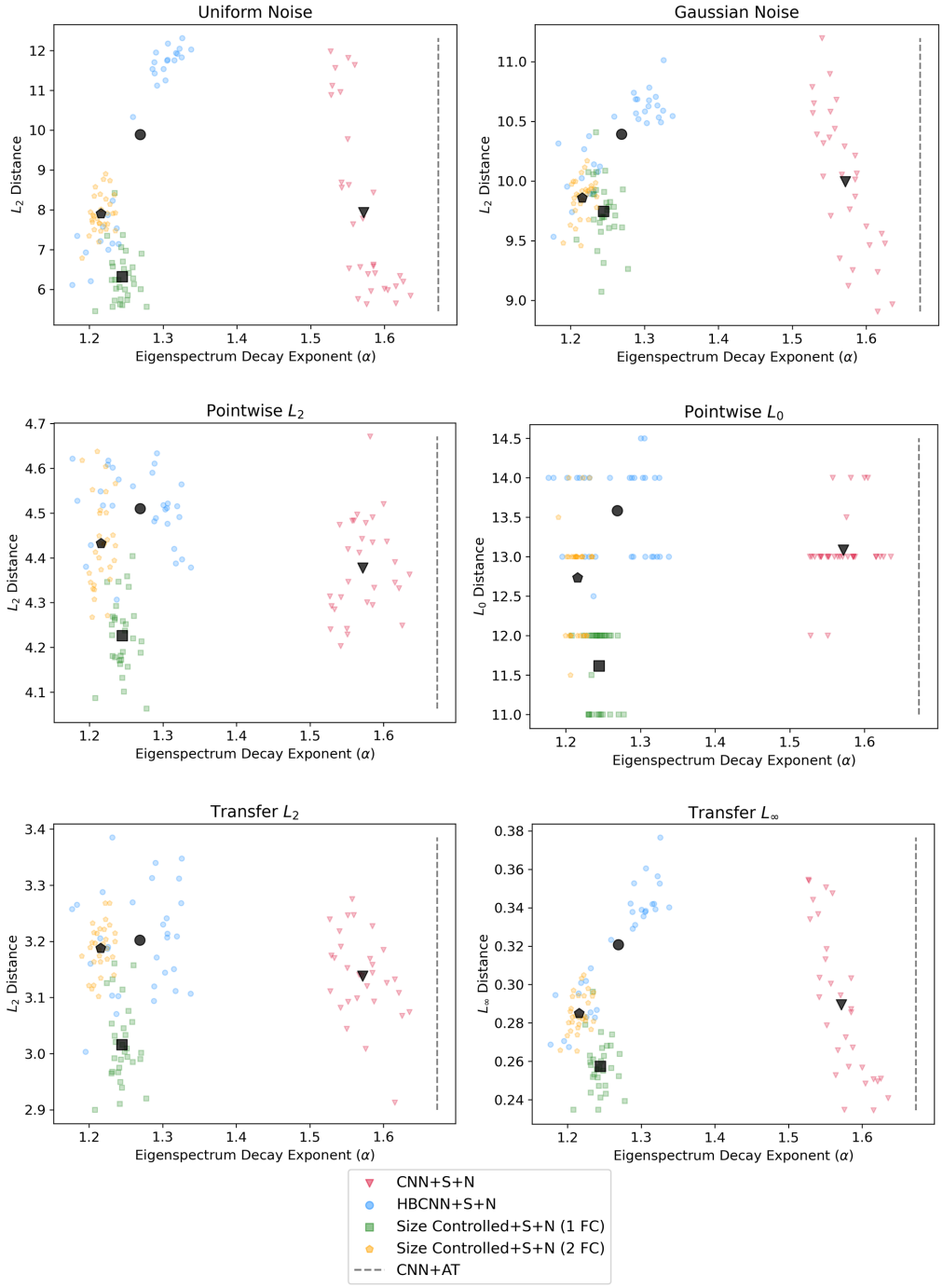


Figure 5.12: *Robustness to Black-Box Attacks vs. α Exponent Fit to Eigenspectrum Decay.* The robustness of individual models as a function of their fitted exponent (α) to the eigenspectrum, where Stringer et al. [2019] observed $\alpha \approx 1$. α is estimated for the layer of activity immediately proceeding the fully connected layers. The black markers indicate the mean value for a given architecture type. The dashed grey-line (far right) indicates the α value of the adversarially trained model from Madry et al. [2018], which is not displayed as a point due to its significant outlier values for robustness. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

potentially more robust) manifold, albeit at a potential cost in the efficiency of the neural code.

5.6.3 Hierarchical Binding and the Restriction of the Adversarial Space

While the above geometric analyses confirmed our opening intuition that hierarchical binding is associated with an expanded neural manifold, and that this contributes to robustness, they still leave open questions about the relationship between hierarchical binding and robustness. In particular, we have argued that a) the binding representations would enable robustness by augmenting the low-dimensional representations already present in a typical CNN architecture, b) these representations are meaningful because they capture the low-level features that causally drive higher-level features, and c) hierarchical binding reduces the number of directions in which the decision boundary is vulnerable to potential attacks. In the following, we provide additional results exploring these claims.

To address (a) and (b), we first perform an analysis where we vary a key hyperparameter, the γ proportion, in our MNIST models. While our main results in the previous section supported point (c), effects from inadvertent gradient masking, which can also affect the successful creation of transfer attacks [Tramèr et al., 2020b], make unequivocally demonstrating this effect challenging. We therefore also perform an analysis with a magnitude-constrained Gaussian attack, demonstrating that hierarchical binding significantly reduces the probability of sampling and identifying a region with an incorrect decision boundary.

5.6.3.1 The Importance of Binding Dimension and Causality

The γ hyperparameter in our model determines the proportion of largest gradients that are used to mask the low-level feature activations that will be up-projected. We implemented this as a means of approximating which of these low-level features were most important in driving the higher representation, with the prediction that the most causally important neurons would be the most useful to augment the other-wise low-dimensional manifold. To explore this, we systematically vary γ, and observe the effect on the robustness of a network. This analysis demonstrates:

- The benefit of increasing the dimension (i.e. number) of low-level binding representations available. More information about the low-level features alongside the abstract, high-level features is generally associated with greater robustness.
- A benefit even without masking (i.e. up-projecting all low-level activations of a layer rather than selecting only a proportion). Thus hierarchical binding's benefit appears to primarily be from the preservation of low-level information alongside high-level features, rather than specifically encoding which low-level features causally drove the high-level representation. The result also provides further evidence against gradient masking as a main effect.
- The benefit of the selected binding representations having played a larger causal role in the representation at the higher level. If only a sub-proportion of low-level features are up-projected, it is generally better to up-project the most causally significant representations.

Figure 5.13 shows that, as the proportion of largest gradients used is increased (using larger values of γ), the network shows a rapid rise in its robustness, presumably as the key causally important dimensions are captured. If γ is too small (e.g. $\gamma = 0.1$), one would expect that not all of the important low-level features are being included and up-projected, and as such there are fewer representations available for informing decision boundaries along the high-dimensional manifold. With an even smaller value ($\gamma = 0.01$), it appears that the representations may be too labile to be useful, and that they can in fact be harmful to robustness. Interestingly, there is a rapid rise as γ increases, but this benefit then largely plateaus. This result suggests that while the primate brain might implement hierarchical binding for a variety of computational purposes, we observe that the primary influence this has on robustness is from the preservation of low-level features alongside more abstract features, rather than the additional encoding of which low-level features causally drove more abstract representations. This finding is consistent with our proposed connection between hierarchical binding, manifold geometry, and robustness, where the preservation of low-level features alone should provide a more expressive neural manifold.

When γ reaches 1.0, we are simply up-projecting all the low-level features alongside those of max-pooling (i.e. without any masking). The benefit in spite of no masking pro-

cedure demonstrates that our main effect cannot be explained by our gradient-unpooling operation introducing discontinuities, as has been found in high-sparsity regimes of k-Winner-Take-All [Xiao et al., 2020]. While we noted above that a higher γ appears useful for several attacks, this is not universally the case, and in particular a higher γ reduces the robustness of the classifier to the Boundary Attack by making the decision boundary more linear and amenable to attack. We tuned γ in our networks for robustness across the range of attacks leveraged (including Gaussian Noise and gradient-based attacks), while capturing the notion of sparseness embodied in our opening, biological motivation. In the case of MNIST, results from the cross-validation data suggested that a value of 0.4 was optimal.

Figure 5.13 also shows that using the same algorithm for generating the gradient unpooling representations, but with the γ-*smallest* gradients to derive the binding information instead, generally confers less benefit until virtually all activations are up-projected. This supports the notion that our implementation of gradient-unpooling is capturing some information about which low-level representations are most useful for supporting robust decision boundaries.

Note that we do not claim that the implementation we use is necessarily the best possible for preventing adversarial attacks, and machine-learning focused approaches might want to deviate from our sparse choice of γ. It may be for example that a network using the largest gradients with $\gamma = 1.0$, or a network using the smallest gradients and $\gamma = 0.9$ might be better for certain attacks, such as the Pointwise attack. Rather we aim to demonstrate that our biologically motivated process of up-projecting a sparse representation of the hierarchical binding features confers robustness, and that this generally performs better when the most important (by our approximate measure) features are up-projected, rather than the least important. The noisiness of the trends in Figure 5.13 may partly relate to the ultimate failure of our gradient unpooling operation to perfectly capture which low-level neurons were causally important to the more abstract representation.

Note that we do not show results for the γ-smallest gradients on the L_∞ transfer attack, for which additional surrogates would be required, nor any results from the L_2 transfer attack, where the trend is noisier, and interestingly, the unpooling layer (here ablated) appears to

play an important role in robustness. Unpooling was ablated in these experiments to remove the confound of any sparsity in the up-projected representation, and the results suggest that *gradient* unpooling can be sufficient for preventing many of the vulnerable regions leveraged in attacks such as Gaussian and uniform noise. We comment again on the significance of this ablation in the next sub-section.

5.6.3.2 Hierarchical Binding Reduces the Probability of Finding Vulnerable Decision Regions

To provide additional evidence that the decision boundary of the model augmented with hierarchical binding is fundamentally more challenging to attack in a black-box setting, i.e. that it is more difficult to identify vulnerable regions, we leverage a magnitude constrained attack with Gaussian noise. Unlike the additive Gaussian Noise Attack used in our main results which iteratively increases the amount of noise until misclassification is successful, we now add noise sampled from a fixed distribution ($\mu = 0$, $\sigma = 0.35$, i.e. Gaussian noise of greater magnitude than used during training). This attack is repeated up to 50,000 times for each image, and in Figure 5.14, we show the proportion of images that have been misclassified as the number of attempts increases.

As we have noted, our method cannot guarantee the absence of vulnerable regions in the decision boundary, however we would expect these regions to be sampled randomly with very low probability. Whether using a method such as a transfer attack, the Pointwise attack, or indeed brute force noise, one is less likely to arrive in these regions for the hierarchical binding model owing to its improved decision boundary. Consistent with this, we observe that it is much harder to find such regions through random sampling in the HBCNN+S+N than in the standard CNN+S+N or size-controlled models (Figure 5.14). It is interesting to note that, consistent with our earlier results, the size-controlled CNNs appear to be more vulnerable than the standard CNN+S+N. This finding supports that naively adding parameters, rather than striving for representations that reflect the manifold of the underlying data, can be harmful for robustness. It is once again noteworthy that this is particularly the case for the shallower size-controlled CNN with only one hidden layer, despite this being a common motif in models assessed [Schott et al., 2019, Nassar et al., 2020] or designed [Madry et al., 2018] for adversarial robustness.

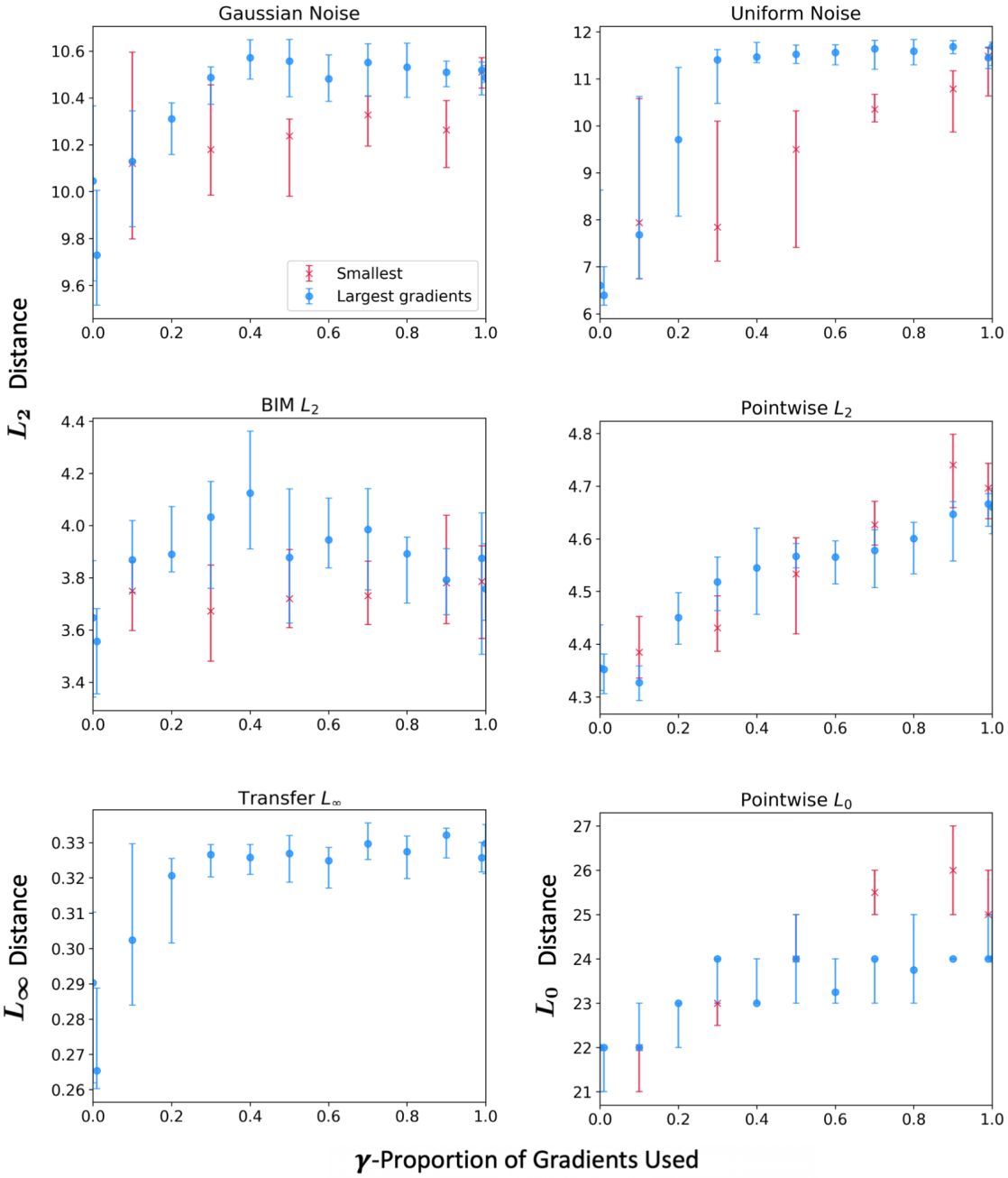


Figure 5.13: *The effect of hierarchical binding dimension and causal role on robustness.* We systematically vary the γ-proportion of gradients used to mask the gradient unpooling representations along the x-axis; results are shown from using both the largest (blue) and smallest (red) γ gradients. 0.0 is equivalent to the standard CNN control model we study, while 1.0 means no masking is applied and all low-level activations are up-projected. The y-axis represents the adversarial perturbation, measured using the appropriate L_p norm for that attack. Each point represents the median distance of a successful adversary, provided as the median performance across 30 networks trained on MNIST without unpooling. Error bars show the 95% confidence interval of the median.

Figure 5.14 also shows the results of two ablations. During and after training, we either always set to 0 the activations of the representations conveying more invariant, abstract information (max-pooling and unpooling, the former after gradient unpooling has been performed), or more low-level information (gradient unpooling). Gradient unpooling appears to provide the majority of the benefit for Gaussian noise leveraged in this manner, while unpooling alone offers little benefit alongside max-pooling. On the other hand, we noted from the ablated networks in the previous section that unpooling appears important for transfer L_2 robustness. We suspect that the directions which unpooling helps the network cover are exceedingly rare to sample randomly, and as such, we see limited benefits in Figure 5.14; indeed we know from our main results that there are *many* vulnerable regions within the space that we sample in this sub-section, and yet clearly these are very difficult to find with brute-force noise.

Finally, the ablations in Figure 5.14 show that gradient unpooling is itself not sufficient to explain the benefit for robustness. If it is not paired with the more abstract representations provided by unpooling and max-pooling, then the network performs worse than the standard CNN+S+N. Once again, this supports our opening motivation of combining these representations for a robust model, rather than gradient-unpooling e.g. introducing an unusual non-linearity that is the source of robustness.

5.7 Results : Performance On Other Data-Sets

To determine whether the observed robustness generalizes to a more complex setting, we apply the same LeNet-5 based architectures to FMNIST, albeit with the gradient-unpooling parameter $\gamma = 0.3$ rather than 0.4, and implement a VGG style HBCNN for the CIFAR-10 data-set with $\gamma = 0.1$ (model details in 5.3.4). Broadly speaking, the model architectures and training protocol for FMNIST is the same as that used for MNIST, including the amount of Gaussian and salt-and-pepper noise. In order to preserve clean classification accuracy on CIFAR-10, our noise-augmented training regime uses Gaussian noise an order of magnitude smaller ($\sigma = 0.03$ vs 0.3), and no salt-and-pepper noise. With the exception of the 'vanilla' model described in the results table, we also use weight decay [Ng, 2004] in the VGG models to further regularize the decision boundaries. We

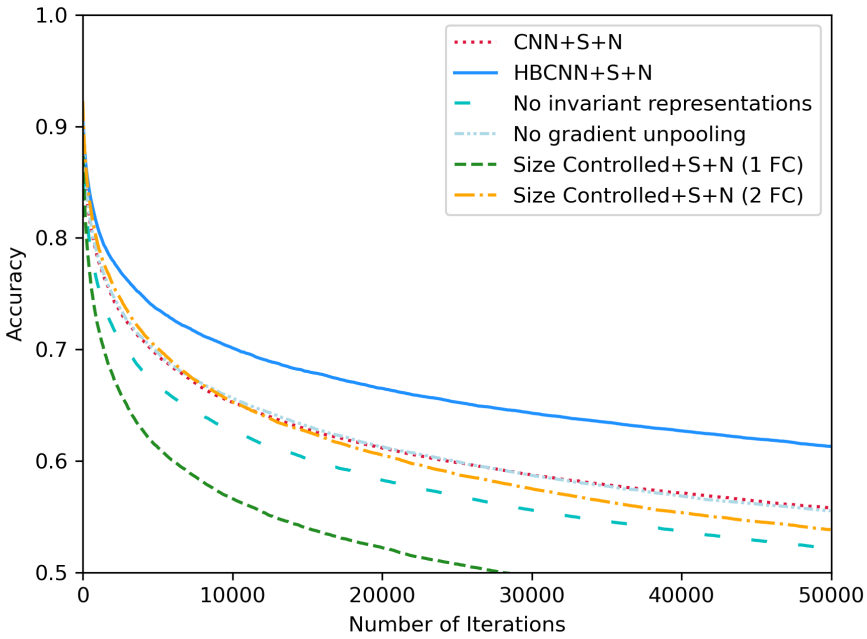


Figure 5.14: *Leveraging repeated Gaussian noise of constant standard deviation.* We show the proportion of correctly classified images across 30 networks as a function of the number of attempted additions of Gaussian noise ($\mu = 0$, $\sigma = 0.35$) to cause misclassification. The 'No invariant representations' model is the HBCNN+S+N model trained and evaluated with the max-pooling and unpooling layers ablated. The 'No gradient unpooling' model is the HBCNN+S+N model trained and evaluated with the gradient-unpooling layer ablated. S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; (n FC) refers to the number of fully connected layers in the size-controlled models.

note that the optimal weight-decay value was tuned for robustness for both the control and the binding-augmented models separately. Given our observation on MNIST that the size-controlled architectures generally resulted in greater vulnerability to adversarial attacks on meaningful metrics, we do not include any in these analyses.

Our results for FMNIST and CIFAR-10 are presented in Tables 5.4 and 5.5 respectively. While there are a few attacks where we fail to generalize the effect of the HBCNN being stronger than the robust control on FMNIST (Gaussian noise, Pointwise L_2, DeepFool attacks, All L_∞) and CIFAR-10 (Pointwise L_2, MIM, All L_0), the trend of enhanced robustness above the control model and across multiple attacks is observed. As such, it appears that some of the benefits of hierarchical binding can generalize to different architectures and data-sets, although the current approach does not result in as strong an effect as we observe in MNIST. Once again, we pair our tabular results with distributions in Figure 5.6 of model performances across several key attacks.

Interestingly, the HBCNN (LeNet-5 variant) surpasses the clean accuracy of the control model on FMNIST (where the control is the 'CNN+S+N' model in Table 5.4). This is not a fair comparison, as the HBCNN has more parameters, but it supports the proposal that when binding representations are included, adversarial robustness becomes a question of on-manifold generalization error, making robustness compatible with clean classification accuracy [Stutz et al., 2019], rather than an orthogonal objective. Finally, we note that on CIFAR-10, while the observed improvement is broad-spectrum, it does not match the highly established method of adversarial training.

5.8 Discussion

We have demonstrated the implementation of a novel CNN architecture, inspired by recent work in theoretical neuroscience [Eguchi et al., 2018]. This architecture approximates hierarchical binding representations by capturing the causal relations between lower-level features and the higher-level representation of an object. Within the framework of adversarial examples as off-manifold perturbations, we presented empirical evidence of enhanced robustness to a variety of attack methods, following the introduction of these

Table 5.4: Fashion-MNIST Results

	Vanilla-CNN	CNN+S+N	HBCNN+S+N	CNN+AT
Clean	90.87%	87.38%	**88.26%**	88.46%
L_2-metric ($\epsilon = 1.5$)				
Transfer ■	1.7 (55%)	3.7 (78%)	**4.2** (81%)	3.1 (80%)
Uniform Noise ■	3.4 (78%)	5.1 (87%)	**5.6** (87%)	3.9 (89%)
Gaussian Noise ■	2.9 (76%)	**8.4** (86%)	8.1 (86%)	3.9 (87%)
Boundary ■	0.4 (7%)	1.6 (52%)	**4.5** (84%)	0.9 (23%)
Pointwise ■	2.3 (70%)	**4.0** (83%)	**4.0** (82%)	1.7 (59%)
FGM	1.2 (45%)	2.5 (62%)	**2.9** (65%)	4.5 (72%)
FGM w/GE	1.2 (44%)	2.5 (62%)	**4.5** (76%)	4.8 (72%)
DeepFool	0.4 (12%)	**1.6** (52%)	**1.6** (52%)	2.3 (66%)
DeepFool w/GE	0.5 (1%)	**2.1** (58%)	1.9 (56%)	2.3 (65%)
BIM	0.3 (1%)	1.0 (36%)	**1.1** (39%)	1.8 (58%)
BIM w/GE	0.3 (8%)	1.0 (35%)	**1.3** (46%)	1.8 (57%)
All L_2	0.3 (0%)	1.0 (32%)	**1.1** (38%)	0.9 (21%)
L_∞-metric ($\epsilon = 0.1$)				
Transfer ■	0.09 (45%)	0.22 (77%)	**0.23** (78%)	0.18 (87%)
FGSM	0.04 (24%)	0.13 (57%)	**0.15** (60%)	0.24 (80%)
FGSM w/GE	0.05 (30%)	0.13 (58)	**0.16** (62%)	0.32 (80%)
DeepFool	0.03 (2%)	**0.13** (58%)	0.12 (56%)	0.26 (79%)
DeepFool w/GE	0.03 (2%)	**0.16** (62%)	0.13 (57%)	0.27 (79%)
BIM	0.02 (1%)	0.07 (37%)	**0.08** (41%)	0.15 (76%)
BIM w/GE	0.02 (9%)	0.07 (36%)	**0.08** (42%)	0.16 (76%)
MIM	0.02 (1%)	0.07 (38%)	**0.08** (41%)	0.15 (76%)
MIM w/GE	0.02 (8%)	0.08 (38%)	**0.09** (46%)	0.16 (76%)
All L_∞	0.02 (0%)	**0.07** (35%)	**0.07** (39%)	0.15 (76%)
L_0-metric ($\epsilon = 12$)				
Pointwise ■	8 (32%)	24 (69%)	**26** (68%)	4 (17%)
Salt&Pepper Noise ■	23 (63%)	**167** (85%)	133 (84%)	16 (54%)
All L_0	8 (32%)	24 (69%)	**26** (68%)	4 (17%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the Fashion-MNIST data-set. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with label smoothing and noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing or noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (LeNet-5 variant); AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/ GE = with Gradient Estimation; ■ = black-box attack.

Table 5.5: CIFAR-10 Results

	Vanilla-VGG	VGG+S+N	HBCNN+S+N	ResNet+AT
Clean	86.51%	**86.43%**	86.07%	87.25%
L_2-metric ($\epsilon = 1.5$)				
Transfer ■	0.68 (26%)	0.99 (36%)	**1.03** (38%)	6.23 (80%)
Uniform Noise ■	3.18 (80%)	5.16 (85%)	**5.40** (85%)	7.64 (86%)
Gaussian Noise ■	2.31 (69%)	3.89 (80%)	**4.12** (79%)	7.50 (82%)
Boundary ■	0.27 (3%)	0.42 (5%)	**4.81** (84%)	1.11 (36%)
Pointwise ■	1.60 (53%)	**1.91** (60%)	1.88 (60%)	2.11 (62%)
FGM	0.26 (21%)	0.39 (26%)	**0.47** (30%)	1.06 (42%)
DeepFool	0.17 (0%)	0.44 (14%)	**0.49** (14%)	0.97 (35%)
BIM	0.14 (1%)	0.22 (1%)	**0.23** (2%)	0.66 (21%)
All L_2	0.14 (0%)	0.22 (0%)	**0.23** (1%)	0.66 (20%)
L_∞-metric ($\epsilon = 8/255$)				
Transfer ■	0.19 (34%)	0.030 (48%)	**0.031** (50%)	0.153 (82%)
FGSM	0.07 (17%)	0.010 (26%)	**0.012** (30%)	0.037 (54%)
DeepFool	0.005 (1%)	0.013 (28%)	**0.015** (28%)	0.039 (56%)
BIM	0.004 (1%)	0.006 (4%)	**0.007** (5%)	0.028 (46%)
MIM	0.004 (1%)	**0.007** (4%)	**0.007** (6%)	0.029 (47%)
All L_∞	0.004 (0%)	0.006 (3%)	**0.007** (5%)	0.028 (46%)
L_0-metric ($\epsilon = 12$)				
Pointwise ■	7 (35%)	9 (42%)	**10** (43%)	11 (45%)
Salt&Pepper Noise ■	15 (53%)	**21** (57%)	**21** (57%)	27 (57%)
All L_0	7 (34%)	**9** (42%)	**9** (43%)	11 (45%)

Results are presented from leveraging a variety of attacks against architectures of interest evaluated on the CIFAR-10 data-set. The main numerical results shown here are the median L_p distances of a successful adversary for the different attacks (rows), provided as the median performance across 30 networks for each model condition (columns). In parentheses is the median accuracy, at a given thresholded perturbation ϵ, taken across 30 networks. The All-L_0, All-L_2 and All-L_∞ distances show the minimal adversarial distance across all attacks of that distance-metric for each image. Bold indicates the best performance between the networks with label smoothing and noise added to the training images; blue indicates the best performance across all networks in both tables. 'Clean' refers to the accuracy on the test data-set without adversarial perturbations; 'Vanilla' indicates the absence of the modifications (regularization with label smoothing and noise in training data) introduced later; HBCNN = Hierarchical Binding-CNN (VGG variant); AT = adversarial training; S = label smoothing; N = Gaussian and Salt-and-pepper noise added to the images during training; w/ GE = with Gradient Estimation; ■ = black-box attack.

representations alongside techniques to promote a robust decision boundary.

In addition to our main attack evaluations, we measured geometric properties of the representations in our HBCNN and standard architectures, observing a relationship between manifold extent and robustness, consistent with our opening premise. Following this, the controlled manipulation of the γ hyperparameter confirmed the value of capturing low-level features that contribute to high-level representations. Finally, the additional analysis of a brute-force noise attack complimented our main results to further demonstrate that the observed robustness aligns with our geometric view of the defence. Further below, we discuss how hierarchical binding representations could help explain the apparent sensitivity of humans to adversarial examples under specific experimental conditions.

One interesting finding of our work is that, while hierarchical binding was implemented as a biologically-inspired defence, the results in Section 5.6.3.1 suggest that the primary observed benefit was due to the preservation of low-level features alongside more abstract, high-level features. As we discussed under our Methods, we designed our architecture to implement a simplified version of the hierarchical binding proposed in Eguchi et al. [2018]; without such constraints, the model would face significant computational challenges owing to the number of features. It may be that future architectures can more faithfully capture this proposed form of hierarchical binding. Under such a setting, the inclusion of hierarchical binding relations might provide additional benefits to robustness beyond the preservation of low-level information achieved by the currently presented model.

A key strength of the proposed approach is the broad range of attack types against which adversarial robustness is enhanced, particularly to black-box attacks across L_0, L_∞ and L_2 metrics. Many adversarial defences result in a trade-off in robustness on one metric (such as to L_∞ attacks) for either minimal robustness to attacks that minimize another distance metric, or even enhanced vulnerability to them [Schott et al., 2019, Laidlaw et al., 2021]. Note for example, the enhanced vulnerability to L_0 attacks seen in the adversarially trained model in Table 5.3. Particularly in the MNIST setting, our architecture appears to defy this typical trade-off, except for low-risk attack settings that require large changes to the input, such as brute-force salt-and-pepper noise. The method also main-

tains much of its accuracy on clean, unmodified images, with these benefits coming at relative computational and parameter efficiency compared to other leading defences on MNIST.

Nevertheless, we must highlight some notable limitations of this work. The proposed architecture adds a hyperparameter (the γ-proportion) that for optimal performance requires tuning, with a general trade-off for different attacks. We found that a more complex dataset appears to benefit from a smaller proportion (MNIST $\gamma = 0.4$, FMNIST $\gamma = 0.3$, CIFAR-10 $\gamma = 0.1$). Furthermore, in the setting of FMNIST and CIFAR-10, the effect size vs. the control was more subtle, and for CIFAR-10, the robustness was not comparable to adversarial training. Finally, the adversarial examples that fool the HBCNN do not appear, on average, to be more meaningful than for the robust control model, and as we discuss below, the model remains vulnerable to a sufficiently capable attacker. Notwithstanding these limitations, our analysis indicates that the hierarchical representations described herein could be an important feature to developing fundamentally robust models. We discuss this in further detail below.

5.8.1 Models That Are Difficult to Fool, vs. Models That Are Impossible to Fool

We have noted that our proposed architecture does not guarantee the absence of vulnerable decision regions, and that a sufficiently powerful attacker may be able to find these. Consistent with this, we have conducted exploratory experiments, requiring re-implementing the architecture in a different deep-learning library (PyTorch), and version of the adversarial attack library (Foolbox 3) in order to leverage newer attacks. While these are provisional results (Foolbox 3 required untested modifications to accept a model such as ours that uses gradients at inference), they suggest that newer attacks such as the Brendel [Brendel et al., 2019] and DDN [Rony et al., 2019] attack may be able to find adversarial sub-spaces that are shared for a standard CNN+S+N and a HBCNN+S+N model, resulting in the same apparent vulnerability given a sufficiently powerful attacker.

This suggests that, given sufficient optimization, the defence can be broken; it may be harder to find vulnerable regions by chance or when leveraging directions effective against

other networks, but in such a high-dimensional space, there will still be many pockets that can be found by a sufficiently powerful attack. This is consistent with the results of Gilmer et al. [2018]. On a synthetic data-set of 500-dimensional spheres, even with enormous training data and sufficient expressive capacity to perfectly separate the class manifolds, a model will learn a decision boundary that significantly deviates from the ground-truth decision boundary, and is therefore vulnerable to adversarial attacks. The intuition is that if the dimensionality is sufficiently high, a model can be 'statistically' perfect (in their case 100% accuracy on 20,000,000 test examples), even if there are *many regions* where the decision boundary extends into the wrong manifold.

In short, we have presented our architecture so as to address an issue of expressive power for learning an appropriate decision boundary, but it cannot fully resolve the inherent challenges of learning these decision boundaries in high-dimensional spaces. As such, low-probability vulnerable regions clearly still exist, which sufficiently powerful attacks are capable of finding. This raises the question of what additional steps could address this persistent challenge. It is possible that unsupervised pre-training [Hénaff et al., 2020, Chen et al., 2020b] would be useful, and indeed techniques based on self-supervised learning have recently been leveraged with success in improving adversarial robustness [Gowal et al., 2021].

As is clear from the results of Gilmer et al. [2018] however, more data alone is unlikely to be sufficient. A complementary approach would be to address how the model generalizes to unseen samples following generic perturbations. Typical neural networks have unfavourable properties in this regard, such as forming polytopes that classify with high probability regions extending into infinity [Hein et al., 2019]. Furthermore, the computations that enable CNNs to achieve translation invariance are likely to contribute to an insensitivity to the spatial configuration of features, and indeed invariance may be contingent on adversarial vulnerability [Kamath et al., 2021]. To achieve more robust generalization, the favourable geometric behaviours of methods such as k-Nearest Neighbour have been noted before as being potentially useful, although these often come at a severe cost in expressive power [Khoury and Hadfield-Menell, 2018, Schott et al., 2019]. Instead, architectures designed to enable robust invariance, such as capsule networks

[Sabour et al., 2017], may be important in combination with the modifications we have presented in this work. As such, the computations we explored in Chapters 3 and 4 could complement the benefits of hierarchical binding, although this remains an unexplored possibility.

5.8.2 Other Methods That Preserve Low-Level Information

It is worth clarifying the relationship of our work to other methods of preserving low-level information. The importance of hierarchical binding for a more robust decision boundary is that the low-level features are represented explicitly in the late stages of the network, with their own activations and thereby unique learned weights projecting to the decision part of the network. Furthermore, this preservation of information is constructed so as to jointly capture a spectrum of feature abstraction. Thus the architecture is designed to avoid more entangled representations of detailed and abstract features as would be found in ResNets [He et al., 2016] and related architectures that can preserve all information about the input, such as i-RevNet [Jacobsen et al., 2018].

In particular, we emphasize that even with $\gamma = 1.0$, our network does not equate to current CNNs with skip connections. In networks such as ResNets, the cross-wise dimension (essentially the spatial resolution of the representation) is progressively decreased throughout the layers of the network through methods such as using convolutions and skip connections with a stride of 2. This results in transforming from a feature map of dimensions such as 56×56 to 7×7 [He et al., 2016]. In our network, the activity is concatenated alongside the representations in the higher layer, rather than added via a skip connection, and as such, the low-level activations maintain their cross-wise dimension (i.e. resolution). Furthermore, these up-projected representations have their own, unique learned weights projecting to the final layers of the network. Similarly, in DenseNets, max-pooling between blocks progressively decreases the feature-map dimension, resulting in lost spatial resolution. Within a block, the channel-wise concatenation of features does not capture the same motif as hierarchical binding of combining low-resolution, abstract features alongside higher resolution, low-level features, as the feature-map dimensions are constant [Huang et al., 2017]. In summary, our architecture represents a

novel method of combining low-level and abstract features, even when $\gamma = 1.0$.

Returning to our initial proposal and Figure 5.1, the simpler read-out of the features that motivates our architecture is important for enabling sufficient representational capacity, as well as amenable features, so as to efficiently learn decision boundaries along all dimensions that constitute class-preserving transformations of the object. The results from our size-controlled CNNs support the importance of this inductive bias built into our architecture - i.e. that low and high-level representations should be represented separately while being conserved up to the decision-level of the network.

In summary, the desire is not to preserve all information in the image through an entangled representation (which deep-invertible networks do perfectly [Jacobsen et al., 2018]), but to promote a robust decision boundary by providing a classifier distinct representations of both the abstract and low-level features of an object. This is thus distinct from the general advantages of skip connections, which can provide benefits such as supporting learning in deeper architectures [He et al., 2016].

5.8.3 Adversarial Examples and Human Perception

Is it possible to connect our biologically-motivated model back to the experimental literature on the robustness of primate vision? Perceptually, adversarial examples in humans might manifest as misclassifying a briefly presented object [Elsayed et al., 2018], or guessing at above chance levels what the adversarial perturbation will induce in machines [Zhou and Firestone, 2019]. Such effects might be achieved by either randomly adding noise in the hope of finding an adversarial example, or leveraging transfer attacks from machine learning models [Elsayed et al., 2018].

Humans without time constraints do not appear to be sensitive to adversarial perturbations (note the limitations identified by Dujmović et al. [2020] regarding the work in Zhou and Firestone [2019]), and as such the only experimental evidence for human sensitivity to adversarial examples is from Elsayed et al. [2018]. They used transfer attacks created from an ensemble of CNNs to determine whether these would impact human accuracy in a two-alternative forced choice task (e.g. cat vs. dog). While the effect on classification was not comparable to the dramatic shift seen in machine vision systems, they measured

statistically significant drops in accuracy when humans were constrained to view adversarial images for a very short duration (around 60-70ms) followed by masking intended to limit recurrent and top-down processing.

The CORNet-S architecture is a CNN variant that makes use of recurrent activity in an attempt to better match the primate ventral stream [Kubilius et al., 2019]. Together with additional modifications in the form of a front-end that corresponds to the processing seen in V1, Dapello et al. [2020] demonstrated its enhanced robustness above a CNN without recurrence. Without this modified front-end however, CORNet-S appears to be more vulnerable than a typical CNN without recurrence, such as AlexNet, to white-box adversarial examples, making it unclear what the specific contribution of recurrence to robustness might be. In addition, the evaluation in Dapello et al. [2020] did not include transfer attacks, the method used in Elsayed et al. [2018]. Similarly, in Huang et al. [2020], the value of recurrent feedback for robustness to adversarial examples was demonstrated, although the evaluation of black-box attack methods such as transfer attacks was limited to versions of their model that had undergone adversarial training.

Given the uncertain role of recurrent computations for robustness based on previous models, how might the effects in Elsayed et al. [2018] be explained? As noted in our introduction, it has been predicted that top-down and lateral activity in a spiking neural network would be needed to implement the proposed hierarchical binding algorithm in a biological system. As such, the disruption of such processing (e.g. via brief presentations and masking) may reduce the quality of hierarchical binding representations, and could thereby provide a possible basis for the observed effect in Elsayed et al. [2018].

More generally, there have been several valuable contributions to explain human robustness to adversarial examples, but we feel these still leave important questions unanswered. Vuyyuru et al. [2020] demonstrated the effect on robustness of non-uniform retinal sampling and varying receptive field sizes with eccentricity, but their effect was specific to small perturbations, and the evaluation of black-box attacks did not assess whether transfer attack robustness was greater for the proposed model vs. an undefended one. Tadros et al. [2020] examined the effect of sleep-like algorithms on robustness, and again while this might account for some of the difference between humans and artificial systems, their

method actually increased the vulnerability of the model on MNIST to the Boundary Attack (see their Table 1), with no evaluation of other black-box methods such as transfer attacks or the Pointwise attack. Nassar et al. [2020] examined whether regularizing representations to follow the $1/n$ power-law observed in Stringer et al. [2019] could explain the benefits of biological neurons displaying such characteristics. While their results suggested that this might indeed partly explain robustness in biological vision, our own results suggest that this also has limited explanatory power across diverse architectures. A similar approach proceeding Nassar et al. [2020] was taken in Li et al. [2019], where the representations of a ResNet model were encouraged to align with responses from the mouse visual system. This promising work demonstrated robustness to both noise and a variety of L_p metrics, although it remained unclear what the specific mechanism was for this robustness. Finally, a recent follow-up analysis of the model introduced by Dapello et al. [2020] assessed the contribution that stochastic neural responses make to adversarial robustness [Dapello et al., 2021]. Interestingly, their results show that stochasticity increases the neural manifold's extent, and that this is associated with robustness (i.e. consistent with our own results). However, their actual analysis focused on the (lack of) change in manifold representations in response to adversarial examples, there was no inclusion of black-box attacks, and we highlight again that this work followed our own. In summary, we believe our work helps address an explanatory gap regarding human robustness to adversarial examples, in particular to transfer attacks - the one method that has been leveraged against humans [Elsayed et al., 2018], and a threat setting where our method consistently enhances robustness.

An additional relevance of our results to the human perception of adversarial examples is the question of a robustness vs. accuracy trade-off. Some theoretical results have suggested that adversarial robustness and accuracy on unmodified, clean data-sets are mutually incompatible objectives and that robustness necessarily comes at a cost to clean accuracy [Tsipras et al., 2019]. This notion, however, is counter-intuitive given the observation that humans are both accurate generally, and robust to adversarial examples. We noted that our method results in either a limited drop (or in the case of Fashion-MNIST improvement) in clean classification accuracy, suggesting that these objectives are not in fact mutually exclusive. More specifically, the modest drop in clean accuracy we see on

MNIST is less than for other robust models, yet our model uses either fewer parameters (cf Madry et al. [2018]), or less compute time (cf Schott et al. [2019]) than these. We have argued that when hierarchical binding captures additional dimensions that describe the true, underlying manifold of the object, a robust decision boundary becomes a case of on-manifold generalization [Gilmer et al., 2018, Stutz et al., 2019], and rapid classification can proceed as normal. In summary, our approach combines robustness (at least to attacks that can be plausibly leveraged against humans), efficiency (both parameter and compute-wise) and persistent clean accuracy. We believe this provides additional evidence that the basis of adversarial examples and robustness that we explore is well aligned with what distinguishes vision in artificial systems and primates.

5.9 Conclusion

The work presented in this chapter provides evidence for the hypothesis that the preservation of low and high-level visual features at different spatial scales is important for robust object recognition, whether through hierarchical binding in the primate brain or in artificial systems. While our strongest results are on MNIST, robustness on this simplest of image data-sets remains far from solved [Schott et al., 2019, Mu and Gilmer, 2019, Tramèr et al., 2020a, Nassar et al., 2020]. Future systems that better preserve hierarchical binding information and make use of it may be important to generalizing the observed robustness to more complex settings, such as the ImageNet [Russakovsky et al., 2015] data-set of natural images.

6 | General Discussion

In this thesis, we have set out to explore the connection between the decades-old "binding problem", and the computational basis of robust vision in both machines and primates. Our work was motivated by the observation that we appear tantalizingly close to explaining many elements of primate vision given current leading models in machine learning. At the same time, these architectures continue to demonstrate brittle behaviour under certain conditions, indicating that they remain imperfect models of biological vision.

One consideration motivating our work was whether recently developed algorithms, designed to address particular robustness issues, might inform computations that could be implemented in spiking neural networks. In such a way, ideas from machine learning might help us better understand visual processing in the brain. Taking the other direction of knowledge transfer, novel theories continue to emerge in the neuroscience community to explain how the brain approaches binding. Current machine vision systems remain vulnerable to a variety of perturbations, even when allowed access to computations, hardware and training data unavailable to biological agents. As such, we set out to adopt inductive biases informed by recent neuroscience theory, and thereby design more robust machine learning systems.

In Chapter 2, we investigated the challenging setting of integrating visual inputs across a series of saccades, as would be necessary for an agent operating in a natural environment. Given the presentation of new stimuli under such settings, a binding mechanism is necessary to enable rapid token creation, and in such a way that the spatial relations of these tokens can be meaningfully understood across saccades. Grid cells have recently been identified as a candidate basis for binding, including in visual space [Whittington et al., 2018, Hawkins et al., 2019, Lewis et al., 2019, Bicanski and Burgess, 2019, Whittington et al., 2020]. We therefore explored the capabilities of a model that used a combination of grid-cell path integration, and rapid, associative binding. In the domain of object recognition, we observed that the model in question was able to robustly recognise objects, even when these are perceived over sequence trajectories that differ at training and evaluation time. Such generalization to novel sequence trajectories demonstrates the capability of

grid cells to support a form of out-of-distribution generalization. More importantly, such capabilities do not come at a loss in sensitivity to the spatial configuration of features. In particular, the conjunction detection that GridCellNet performs makes use of the information provided in the spatial arrangement of features, ensuring that nonsense objects with no semantic meaning to humans are not "recognised" as objects.

In addition to evaluating GridCellNet in the domain of object recognition across a series of saccades, we demonstrated that the rapid token creation it performed is useful for a variety of tasks. For example, specific objects that were seen during training time can be recalled, and GridCellNet can predict unsensed regions of the input space, enabling the reconstruction of partially observed objects (whether these are familiar or novel examples). While all performance was assessed in the challenging context of few-shot learning, we also identified that GridCellNet's use of sparse representations and Hebbian learning rules enabled robustness in a continual learning setting.

In summary, grid cells enable tokens that leverage internal reference frames, and with path integration capabilities. Our results suggest that such a representational format can serve as a powerful substrate for object representations, enabling strong performance in a series of challenging task domains. Furthermore, the structure of GridCellNet is broadly consistent with the known neuroanatomy of the primate brain. As such, it may also prove useful as a model of primate vision.

In Chapters 3 and 4, we turned to the question of how robust conjunction detection can be performed in parallel, across the visual stream. Since the identification of complex cells by Hubel and Wiesel [1962], such processing has typically made use of pooling operations to reduce sensitivity to the spatial particularities of features, with the hope that this can support invariance to online translations, or in other words, novel locations in the input space. An alternative approach that was adopted both in machine learning models of object recognition [Sabour et al., 2017], and computational neuroscience models of word recognition [Davis, 2010b], considered the possibility that additional information could be captured in the response profiles of connectionist systems, such as in the time with which neurons spike. Such category slots can be responsive over a large receptive field, whilst encoding additional spatial information, thereby enabling downstream conjunction

detection that is both translation invariant, and sensitive to the precise arrangement of features.

Building on the prior conceptual work of Davis [2010b] and Sabour et al. [2017], we set out to determine whether biologically plausible spiking neural networks could use STDP-learning alone to leverage spatial information encoded in the relative spike times of feature neurons. We performed a series of analyses and simulations, demonstrating that such computations are possible, whilst also allowing us to characterise the parameter space within which they might feasibly operate. Furthermore, we highlighted the complementary role that such computations might play alongside typical, pooling-based solutions to invariant conjunction detection. Whilst these models address the robustness task that motivated the work, a variety of limitations suggest that additional learning mechanisms, such as myelin plasticity [Bowers, 2017, Bowers and Davis, 2017] are likely to be necessary for such a form of conjunction detection to be widely implemented in the primate brain.

Finally, in Chapter 5, we examined the significance of hard-coded, hierarchical binding representations for the phenomenon of adversarial examples. Such hierarchical binding representations have been predicted in the primate visual system, where it is proposed that these would help preserve low-level information about visual objects [Eguchi et al., 2018]. Hierarchical binding would significantly enrich the representations available to the primate visual stream, and in particular expand the neural manifold associated with an object. Drawing on evidence that adversarial examples leave low-dimensional data manifolds, whilst remaining largely unchanged to humans, we developed a novel CNN architecture to approximate the encoding of hierarchical binding, and thereby enhance robustness to adversarial examples.

Our results demonstrated that adversarial examples are indeed more challenging to leverage against the modified architecture, particularly in the context of black-box adversarial attacks, which are constrained in their ability to search the vulnerable space of a model's decision boundary. Further analyses confirmed that these changes are related to the geometry of a network's representations, supporting the initial hypothesis of our work. We further noted the possible connection between persistent vulnerability to sufficiently pow-

erful attacks, and the challenges of decision boundaries in high dimensional spaces. Finally, we considered the possible significance of our architecture for understanding the mild sensitivity that humans can display to adversarial examples under appropriate experimental conditions [Elsayed et al., 2018].

6.1 Interactions Between the Explored Computations

We now consider how the above approaches to binding and robustness might interact with one another. GridCellNet implements a mechanism to rapidly bind sensory features in an internal reference frame, where these features can be accumulated across multiple fixations. The sensory features it receives, however, are assumed to have passed through a parallel processing stream. If the brain makes use of the spatially-sensitive conjunction detection explored in Chapters 3 and 4, then associative, Hebbian binding could take place between these conjunction cells and the grid cells. In addition to these conjunction cells, the hierarchical binding computations we explored would increase the dimensionality of the representations processed at each fixation. While it is unclear how much of this detail would be written into the associative memory with grid cells [Nie et al., 2017], this change in the neural manifold might still have consequences for processing. What might the computational significance of such interactions be?

Adversarial examples are arguably the most confounding problem to suggestions that current deep-learning models represent strong models of primate vision, while the engineering of systems that are robust to them has represented a significant, albeit largely unsuccessful endeavour for nearly a decade [Tramèr et al., 2020b, Croce and Hein, 2020]. Similarly, leading models in machine vision have less of a bias towards object shape than humans, while shape bias is associated with adversarial robustness [Geirhos et al., 2021]. Combining the computations we consider might play some role in addressing these challenges.

More specifically, there is evidence that foveal (i.e. non-uniform) processing performed by the primate retina might confer some robustness to adversarial examples [Vuyyuru et al., 2020]. To address both adversarial robustness and confer greater shape bias, it may

be necessary that object processing involves a combination of spatially-sensitive conjunction detection *across* saccades (i.e. through a network such as GridCellNet), foveal sampling of the input (i.e. as in [Vuyyuru et al., 2020]), and finally spatially-sensitive conjunction detection *within* each foveation (i.e. through a mechanism such as was explored in our spiking neural networks). Hierarchical binding could also play a role in enhancing adversarial robustness. In particular, we identified evidence that the resulting higher dimensional representations cause many adversarial attacks to fail. At the same time, prior analyses of neural network decision boundaries [Gilmer et al., 2018] suggest that poor generalization to unsampled data points is also a key basis of adversarial examples, not just the dimensionality of representations. If spatially sensitive conjunction detection generalizes better to un-sampled representational space, it may prove complementary to hierarchical binding, and thereby further improve the robustness of representations within each fixation.

Taken together, combining these four aspects (i.e. the three computations we explored, and the finding from Vuyyuru et al. [2020]), might represent an approach to both enhancing shape bias in machine vision models, as well as reducing (at least to some degree) vulnerability to adversarial examples.

From a neuroscience perspective, we have noted that current CNN models and their variants are able to account for a significant amount of variance in the primate ventral stream, and yet there remains an explanatory gap [Yamins et al., 2014, Bashivan et al., 2019]. Importantly, the computations we have proposed for spatially sensitive conjunction detection and hierarchical binding do not replace the computations that these CNNs implement, but rather augment them. It may be that models that draw on all of these computations would help to address the current gap in explaining the variance of neural responses in primate visual cortex, although this is not something that we have explored in our work.

6.2 Where is the Solution to Binding Likely to be Found?

In this thesis, we have focused on the potential role of grid cells for rapidly binding features across visual saccades, as well as hard-coded binding representations that are

based on delay-line computations. Naturally, however, there are many other exciting developments in machine learning and neuroscience that may ultimately represent the correct approach to a general description of binding, both as implemented in the brain, and for the design of robust machine learning systems.

In the machine learning literature, two particularly prominent trends have emerged in recent years, which are the power of transformer architectures [Vaswani et al., 2017], and the importance of scaling model and data-set sizes. Continuing to focus our discussion on the problem of vision, the former trend has manifested in the impressive performance of visual transformer (ViT) architectures in a variety of challenging task domains [Dosovitskiy et al., 2021]. Beyond simply demonstrating enhanced object recognition, ViTs display the ability to perform challenging tasks including predicting the occluded region of complex, novel images [Chen et al., 2020a]. There is also some evidence that the representations of ViTs more closely align with human perception than CNNs [Tuli et al., 2021, Geirhos et al., 2021]. Particularly interesting, transformer architectures form a key component in recent models trained to generate embeddings that correctly pair language and visual inputs. Such "CLIP"-based training might involve, for example comparing the representation derived from the description "Pepper the aussie pup" with that derived from a particular image of a dog. Following extensive training with such input pairs, CLIP-models are able to perform zero-shot classification on novel object classification tasks, simply by providing a text query such as "A photo of a (blank)" [Radford et al., 2021]. As well as these remarkable generalization capabilities, CLIP-based embeddings have recently been used to generate photo-realistic images that require the composition of abstract features [Ramesh et al., 2022]. Demonstrative examples of such outputs are provided in Figure 6.1. Given the inherently compositional nature of these generated images, it seems clear that at some level, the representational embeddings in these models have arrived at a workable solution to arbitrary feature binding. Such results demonstrate the profound advances that have been made in the field, with likely significant implications for society.

In many respects, the success of transformer networks in binding is not surprising. Their use of positional-encoding is relatively flexible (although see our earlier discussion in

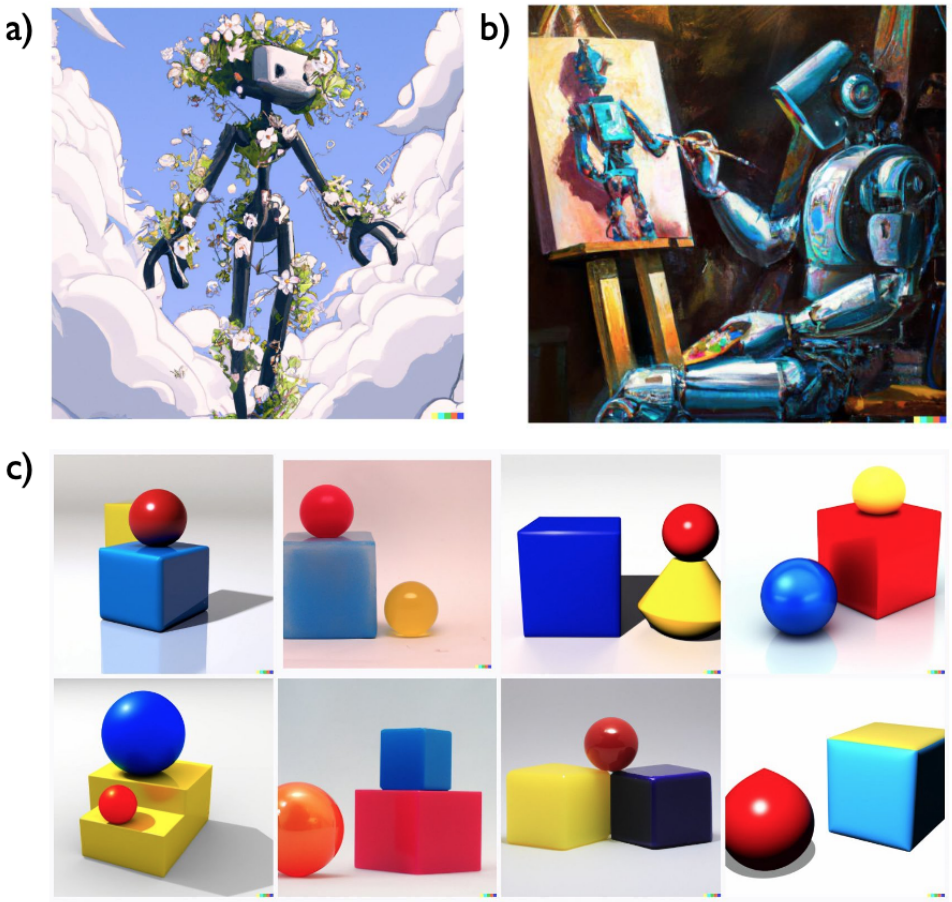


Figure 6.1: *A series of images generated by a state-of-the-art machine learning model.* Shown are a series of reproduced images from a recently published machine-learning model [Ramesh et al., 2022] demonstrating that some complex forms of binding appear well within the capabilities of such models, while relatively simple aspects of binding remain challenging. Provided in quotation marks is the prompt that was given to the model. Note that this is the only information that the trained model receives; it must then *generate* a 1024×1024 pixel image that would correspond to this description. a) "A happy robot with flowers growing out of the top of its head with clouds in the background, digital art". b) "A dramatically lit brightly colored detailed painting of a robot artist painting a self-portrait". c) A random selection of the model outputs when given the prompt, "A blue cube, on top of a red cube, beside a smaller yellow sphere". In the case of (c), none of the generated images fully match the description, although some are close. The images in (a) and (b) are from the open-source art collection "1111101000 Robots" [Barry, 2022]; the images in (c) were generated by, and are reproduced with the permission of, D. Madras, personal communication, April 2022.

Chapter 2), while the self-attention computations are inherently relational. In spite of exciting advances however, it is not clear that transformers will enable a complete solution to binding. Current evidence is that visual transformers remain as vulnerable to adversarial examples as CNNs, given appropriate comparisons [Bai et al., 2021, Fu et al., 2022]. Whether visual transformers more accurately reflect human biases (such as to shape) than CNN architectures appears to reflect, to a large degree, the data-sets and training objectives on which they are typically trained [Bhojanapalli et al., 2021], rather than a fundamental difference in the computations they perform. From a neuroscience perspective, it is likely that the self-attention operation has a rough biological correlate [Mumford, 2020, Bricken and Pehlevan, 2021], yet it remains unclear how the brain would apply this in a massively parallel fashion using shared weights, key components underpinning the power of transformers. Finally, in spite of the often astounding performance of models such as those presented in Ramesh et al. [2022], they still display unusual quirks. For example, the generated images often show inconsistencies in low-level details across space, such as missing limbs or eyes. Moreover, composition appears to frequently fail when confusion can arise about the binding relations in an image, or in other words, the classic superposition catastrophe. We include an example of one such failure in Figure 6.1c. This example is notable for its apparent simplicity, requiring the binding of colours to geometric objects. On the one hand, architectures that leverage transformers are able to solve extremely challenging forms of binding, such as appropriately melding the abstract concepts of "painting", "robot", and "self-portrait" into a coherent whole. On the other hand, the way in which these binding operations are performed still lacks robustness and generalizability.

An additional (sometimes overlapping) trend that has emerged in recent years is the power of scaling both model size (i.e. number of learnable parameters), and data-set size. Many of the top-performing architectures now have hundreds of billions of parameters, with systematic differences observed in model performance as size increases [Brown et al., 2020]. Across a variety of architectures, it appears that increasing the size of the data-set on which a machine vision model is trained by 1-3 orders of magnitude results in significantly more human-like processing [Geirhos et al., 2021]. Transformer architectures are very flexible in the representations they can learn, and it may be that, given sufficient data

and parameters, such an architecture can naturally find a relatively general solution to binding through optimization alone. Since our own brains model the statistical structure of the world, any such solution might be relatively similar to the biological approach that emerges through millions of years of evolution, followed by years of embodied learning.

Leveraging data to improve the alignment of models to human processing may not necessarily be sufficient however. For example, adversarially trained models show a greater shape bias, but such training also results in models demonstrating *worse* robustness to many random image corruptions [Schott et al., 2019, Geirhos et al., 2021]. More importantly, humans are able to demonstrate robustness to adversarial examples without the highly unnatural (and computationally expensive) process of adversarial training. Our intuition therefore is that data and model-scaling approaches may arrive at binding solutions that do not align exactly with those operating in the brain, and more importantly, may continue to harbour limitations that our own neural architectures do not.

By analogy, current aircraft deviate significantly from biological flight, using fixed wings rather than the moving wings that characterise ornithopters. It is not too surprising that the first successful implementations of mechanical flight deviated from biology, as aircraft are designed to carry (often hundreds of) human beings over large distances, and at incredible speeds. Such tasks, however, take less consideration for constraints such as energy consumption or sound production, and they do not require implementation in machines as small as 1mm (such as the insects of the family Mymaridae, or "fairyflies" [Corbett and Rosenheim, 1996]). By contrast, biological flight is constrained to carry small loads in a way that is energy efficient, robust to mid-air collisions, and often silent so as to avoid detection. Consistent with this dichotomy in optimization settings, recent work has demonstrated the comparative energy efficiency of butterfly flight [Johansson and Henningsson, 2021], the benefits of insect-like actuators for surviving mid-air collisions [Chen et al., 2021b], and the utility of owl-like wing serrations in reducing aircraft noise [Rao et al., 2017].

In other words, it is likely that today's approaches of scaling highly expressive models on ever larger data-sets may arrive at a relatively flexible and general solution to binding. However, it is our view that an approach which carries the hallmarks of human intelli-

gence will still benefit from the brain as a roadmap. For example, learning from small amounts of data is not only useful, but it may also be important for preventing models from internalizing harmful (e.g. racist or sexist) information, a significant concern with current approaches [Brown et al., 2020, Radford et al., 2021]. In particular, maintaining oversight into the internalized biases of a model is challenging when data-sets can be as large as 400 million pairs of images and text gathered from the internet [Radford et al., 2021]. Similarly, energy efficiency is a clear strength of the human brain, which is generally estimated to consume less than 20 watts of energy [Sokoloff, 1991]. By contrast, a recent state-of-the-art model was estimated to require more than 188,000 kWh of energy for training [Anthony et al., 2020] (or by our estimate, just over 1,000 years of human brain activity). Without similar constraints, it is likely that the solutions arrived at in machine learning models will bear only some resemblance to the computations performed in our own brains.

Turning to biology, one nascent area in theoretical neuroscience which we believe to be particularly promising is the potential role of grid cells in binding. While we naturally explored this in our own work, we note that this was still through a fairly narrow view. In particular, grid cells are capable of modelling space that need not be limited to 2D grids [Whittington et al., 2020], which has interesting implications for how our brain might represent the relations of object parts. As noted in our main introduction, there is also early work looking at how grid cells might support hierarchical binding, encoding the relations of features across multiple scales of abstraction [Whittington et al., 2018, Hawkins et al., 2019, Whittington et al., 2020]. This is an interesting prospect given the challenges that we have outlined in performing hierarchical binding for arbitrary feature relations.

In addition to grid cells however, it has recently been argued that a key contributor to poor generalization in current machine vision systems is the absence of relational information as processed in the parietal cortex [Summerfield et al., 2020]. Given the role of parietal cortex in relational reasoning, such as the concepts of "larger than", and "to the left of" [Friedman-Hill et al., 1995], as well as its clear significance for visual feature binding [Chechlacz and Humphreys, 2014], we look forward to future work investigating

the importance of such computations for enabling robustness and generalization.

In summary, both in the theoretical work that we have touched upon, as well as other ideas being developed in machine learning and computational neuroscience, there exist a variety of promising paths towards understanding the basis of binding. Regardless of the ultimate description that is arrived at, it is clear that this solution is likely to be multi-faceted, and with profound implications for both the understanding of our own brains in health and disease, as well as the design of machine intelligence.

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 2016. ISBN 9781931971331.

N. Ahmad, M. van Gerven, and L. Ambrogioni. GAIT-prop: A biologically plausible learning rule derived from backpropagation of error. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

S. Ahmad and L. Scheinkman. How Can We Be So Dense? The Robustness of Highly Sparse Representations. *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

S. M. Ali Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 2018. ISSN 10959203. doi: 10.1126/science.aar6170.

R. A. Andersen, G. K. Essick, and R. M. Siegel. Encoding of spatial location by posterior parietal neurons. *Science*, 230(4724), 1985. ISSN 00368075. doi: 10.1126/science.4048942.

J. Anderson, I. Lampl, I. Reichova, M. Carandini, and D. Ferster. Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex. *Nature Neuroscience*, 3(6), 2000. ISSN 10976256. doi: 10.1038/75797.

M. C. Angulo, J. Rossier, and E. Audinat. Postsynaptic glutamate receptors and integrative properties of fast- spiking interneurons in the rat neocortex. *Journal of Neurophysiology*, 82(3), 1999. ISSN 00223077. doi: 10.1152/jn.1999.82.3.1295.

L. Anthony, B. Kanding, and R. Selvan. Carbontracker: Tracking and Predicting the

Carbon Footprint of Training Deep Learning Models. *ICML Workshop on "Challenges in Deploying and monitor- ing Machine Learning Systems"*, 2020.

S. D. Antic, W. L. Zhou, A. R. Moore, S. M. Short, and K. D. Ikonomu. The decade of the dendritic NMDA spike, 2010. ISSN 03604012.

N. Asghar, L. Mou, K. A. Selby, K. D. Pantasdo, P. Poupart, and X. Jiang. Progressive memory banks for incremental domain adaptation, 2020. ISSN 23318422.

F. G. Ashby, W. Prinzmetal, R. Ivry, and W. T. Maddox. A formal theory of feature binding in object perception. *Psychological review*, 1996. ISSN 0033-295X. doi: 10.1037//0033-295X.103.1.165.

A. Athalye, N. Carlini, and D. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *35th International Conference on Machine Learning*. MIT, 2018.

A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20, 2019. ISSN 15337928.

C. M. Bacmeister, R. Huang, M. A. Thornton, L. Conant, A. Chavez, A. Poleg-Polsky, and E. G. Hughes. Motor Learning Drives Dynamic Patterns of Intermittent Myelination on Behaviorally-Activated Axons. *bioRxiv*, 2021.

V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2017. ISSN 01628828. doi: 10.1109/TPAMI.2016.2644615.

Y. Bai, J. Mei, A. Yuille, and C. Xie. Are Transformers More Robust Than CNNs? *35th Conference on Neural Information Processing Systems (NeurIPS 2021).*, 2021.

N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology*, 2018. ISSN 15537358. doi: 10.1371/journal.pcbi.1006613.

S. N. Baker and R. N. Lemon. Precise spatiotemporal repeating patterns in monkey pri-

mary and supplementary motor areas occur at chance levels. *Journal of Neurophysiology*, 84(4), 2000. ISSN 00223077. doi: 10.1152/jn.2000.84.4.1770.

M. D. Barense, T. J. Bussey, A. C. Lee, T. T. Rogers, R. R. Davies, L. M. Saksida, E. A. Murray, and K. S. Graham. Functional specialization in the human medial temporal lobe. *Journal of Neuroscience*, 25(44), 2005. ISSN 02706474. doi: 10.1523/JNEUROSCI.2704-05.2005.

P. Barham and M. Isard. Machine Learning Systems are Stuck in a Rut. In *Proceedings of the Workshop on Hot Topics in Operating Systems, HotOS 2019*, 2019. doi: 10.1145/3317550.3321441.

B. Barry. *1111101000 Robots*. 2022.

J. J. Barton, G. L. Malcolm, and R. L. Hefter. Spatial processing in Bálint syndrome and prosopagnosia: A study of three patients. *Journal of Neuro-Ophthalmology*, 27(4), 2007. ISSN 10708022. doi: 10.1097/WNO.0b013e31815b9b85.

P. Bashivan, K. Kar, and J. J. DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439), 2019. ISSN 10959203. doi: 10.1126/science.aav9436.

D. M. Bear, C. Fan, D. Mrowca, Y. Li, S. Alter, A. Nayebi, J. Schwartz, L. Fei-Fei, J. Wu, J. B. Tenenbaum, and D. L. Yamins. Learning physical graph representations from visual scenes. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. ISSN 01628828. doi: 10.1109/TPAMI.2013.50.

M. J. Berry, D. K. Warland, and M. Meister. The structure and precision of retinal spike trains. *Proceedings of the National Academy of Sciences*, 1997. ISSN 0027-8424. doi: 10.1073/pnas.94.10.5411.

S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit. Understanding Robustness of Transformers for Image Classification. *International Conference on Computer Vision 2021*, 2021.

G. Q. Bi and M. M. Poo. Synaptic modification by correlated activity: Hebb's postulate revisited, 2001. ISSN 0147006X.

A. Bicanski and N. Burgess. A Computational Model of Visual Recognition Memory via Grid Cells. *Current Biology*, 2019. ISSN 09609822. doi: 10.1016/j.cub.2019.01.077.

E. Bienenstock. A model of neocortex. *Network: Computation in Neural Systems*, 1995. ISSN 0954898X. doi: 10.1088/0954-898X{_}6{_}2{_}004.

V. Biscione and J. S. Bowers. Convolutional neural networks are not invariant to translation, but they can learn to be. *Journal of Machine Learning Research*, 22, 2021. ISSN 15337928.

R. Blything, V. Biscione, I. I. Vankov, C. J. Ludwig, and J. S. Bowers. The human visual system and CNNs can both support robust online translation tolerance following extreme displacements. *Journal of Vision*, 21(2), 2021. ISSN 15347362. doi: 10.1167/jov.21.2.9.

R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen. The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks. *Psychological Review*, 113(4):700–765, 2006. ISSN 0033295X. doi: 10.1037/0033-295X.113.4.700.

J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6), 2017. ISSN 15523098. doi: 10.1109/TRO.2017.2721939.

M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, and J. Zhao. End to End Learning for Self-Driving Cars. *arXiv*, arXiv:1604.07316, 2016.

A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 2013. ISSN 01628828. doi: 10.1109/TPAMI.2012.89.

M. M. Botvinick and D. C. Plaut. Short-term memory for serial order: A recurrent neural

network model. *Psychological Review*, 113(2), 2006. ISSN 0033295X. doi: 10.1037/0033-295X.113.2.201.

J. Bowers and C. Davis. Myelin as a medium for plasticity and computation. *Unpublished manuscript*, 2017.

J. S. Bowers. Parallel Distributed Processing Theory in the Age of Deep Networks. *Trends in Cognitive Sciences*, 21(12), 2017. ISSN 1879307X. doi: 10.1016/j.tics.2017.09.013.

J. S. Bowers, I. I. Vankov, M. F. Damian, and C. J. Davis. Neural networks learn highly selective representations in order to overcome the superposition catastrophe. *Psychological Review*, 121(2), 2014. ISSN 0033295X. doi: 10.1037/a0035943.

J. S. Bowers, I. I. Vankov, and C. J. Ludwig. The visual system supports online translation invariance for object identification. *Psychonomic Bulletin and Review*, 23(2), 2016. ISSN 15315320. doi: 10.3758/s13423-015-0916-2.

T. F. Brady, T. Konkle, G. A. Alvarez, and A. Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences of the United States of America*, 105(38), 2008. ISSN 00278424. doi: 10.1073/pnas.0803390105.

T. F. Brady, T. Konkle, and G. A. Alvarez. A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of Vision*, 11(5), 2011. ISSN 1534-7362. doi: 10.1167/11.5.4.

W. Brendel and M. Bethge. Approximating CNNs with Bag-of-Local-Features Models Works Surprisingly Well on ImageNet. In *International Conference on Learning Representations*, 2019.

W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

W. Brendel, J. Rauber, M. Kümmerer, I. Ustyuzhaninov, and M. Bethge. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

T. Bricken and C. Pehlevan. Attention Approximates Sparse Distributed Memory. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

S. L. Brincat and C. E. Connor. Underlying principles of visual shape selectivity in posterior inferotemporal cortex. *Nature Neuroscience*, 7(8), 2004. ISSN 10976256. doi: 10.1038/nn1278.

B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bulthoff, and C. Wallraven. Active in-hand object recognition on a humanoid robot. *IEEE Transactions on Robotics*, 30(5), 2014. ISSN 15523098. doi: 10.1109/TRO.2014.2328779.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

M. J. Buckley, M. C. Booth, E. T. Rolls, and D. Gaffan. Selective perceptual impairments after perirhinal cortex ablation. *Journal of Neuroscience*, 21(24), 2001. ISSN 02706474. doi: 10.1523/jneurosci.21-24-09824.2001.

C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv preprint arXiv:1901.11390*, 1 2019. URL http://arxiv.org/abs/1901.11390.

G. Buzsáki and K. Mizuseki. The log-dynamic brain: How skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4), 2014. ISSN 14710048. doi: 10.1038/nrn3687.

J. A. Cardin, R. D. Kumbhani, D. Contreras, and L. A. Palmer. Cellular mechanisms of temporal sensitivity in visual cortex neurons. *Journal of Neuroscience*, 30(10), 2010. ISSN 02706474. doi: 10.1523/JNEUROSCI.5279-09.2010.

N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfel-

low, A. Madry, and A. Kurakin. On Evaluating Adversarial Robustness. *arXiv*, arXiv:1902.06705, 2019. ISSN 23318422.

C. E. Carr and M. Konishi. Axonal delay lines for time measurement in the owl's brainstem. *Proceedings of the National Academy of Sciences of the United States of America*, 85(21), 1988. ISSN 00278424. doi: 10.1073/pnas.85.21.8311.

M. V. Chafee, B. B. Averbeck, and D. A. Crowe. Representing spatial relationships in posterior parietal cortex: Single neurons code object-referenced position. *Cerebral Cortex*, 17(12), 2007. ISSN 10473211. doi: 10.1093/cercor/bhm017.

M. Chechlacz and G. W. Humphreys. The enigma of Bálint's syndrome: Neural substrates and cognitive deficits. *Frontiers in Human Neuroscience*, 8(MAR), 2014. ISSN 16625161. doi: 10.3389/fnhum.2014.00123.

M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *37th International Conference on Machine Learning, ICML 2020*, volume PartF168147-3, 2020a.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *Proceedings of the 37th International Conference on Machine Learning*, 2 2020b. URL http://arxiv.org/abs/2002.05709.

T. Chen, Z. Zhang, S. Liu, S. Chang, Z. Wang, and T. Ima. Robust Overfitting May Be Mitigated By Properly Learned Smoothing. *ICLR*, 2021a.

Y. F. Chen, S. Xu, Z. Ren, and P. Chirarattananon. Collision resilient insect-scale soft-actuated aerial robots with high agility. *IEEE Transactions on Robotics*, 37(5), 2021b. ISSN 19410468. doi: 10.1109/TRO.2021.3053647.

B. Cheung, J. A. Livezey, A. K. Bansal, and B. A. Olshausen. Discovering hidden factors of variation in deep networks. In *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, 2015.

S. Chung, D. D. Lee, and H. Sompolinsky. Classification and Geometry of General Perceptual Manifolds. *Physical Review X*, 8(3), 2018. ISSN 21603308. doi: 10.1103/PhysRevX.8.031003.

A. Cohen and R. Ivry. Illusory Conjunctions Inside and Outside the Focus of Attention. *Journal of Experimental Psychology: Human Perception and Performance*, 1989. ISSN 00961523. doi: 10.1037/0096-1523.15.4.650.

U. Cohen, S. Y. Chung, D. D. Lee, and H. Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11(1), 2020. ISSN 20411723. doi: 10.1038/s41467-020-14578-5.

R. Condit. Spatial patterns in the distribution of tropical tree species. *Science*, 288(5470), 2000. ISSN 00368075. doi: 10.1126/science.288.5470.1414.

A. Corbett and J. A. Rosenheim. Quantifying movement of a minute parasitoid, Anagrus epos (Hymenoptera: Mymaridae), using fluorescent dust marking and recapture. *Biological Control*, 6(1), 1996. ISSN 10499644. doi: 10.1006/bcon.1996.0005.

M. Corbetta and G. L. Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3), 2002. ISSN 14710048. doi: 10.1038/nrn755.

D. D. Cox and J. J. DiCarlo. Does learned shape selectivity in inferior temporal cortex automatically generalize across retinal position? *Journal of Neuroscience*, 28(40), 2008. ISSN 02706474. doi: 10.1523/JNEUROSCI.2142-08.2008.

F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *Proceedings of the 37th International Conference on Machine Learning*, 2020. ISSN 23318422.

F. Croce, M. Andriushchenko, and M. Hein. Provable robustness of relu networks via maximization of linear regions. In *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics*, 2020.

J. Dapello, T. Marques, M. Schrimpf, F. Geiger, D. Cox, and J. DiCarlo. Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations. *34th Conference on Neural Information Processing Systems*, 2020.

J. Dapello, J. Feather, H. Le, T. Marques, D. Cox, J. McDermott, J. DiCarlo, and S. Chung.

Neural Population Geometry Reveals the Role of Stochasticity in Robust Perception. *35th Conference on Neural Information Processing Systems*, 2021.

C. J. Davis. *The self-organising lexical acquisition and recognition (SOLAR) model of visual word recognition*. PhD thesis, University of New South Wales, 1999.

C. J. Davis. SOLAR versus SERIOL revisited. *European Journal of Cognitive Psychology*, 22(5), 2010a. ISSN 09541446. doi: 10.1080/09541440903155682.

C. J. Davis. The spatial coding model of visual word identification. *Psychological Review*, 117(3), 2010b. ISSN 0033295X. doi: 10.1037/a0019738.

C. J. Davis and J. S. Bowers. What do letter migration errors reveal about letter position coding in visual word recognition? *Journal of Experimental Psychology: Human Perception and Performance*, 30(5), 2004. ISSN 00961523. doi: 10.1037/0096-1523.30.5.923.

O. de Faria, H. Pivonkova, B. Varga, S. Timmler, K. A. Evans, and R. T. Káradóttir. Periods of synchronized myelin changes shape brain function and plasticity. *Nature Neuroscience*, 24(11), 2021. ISSN 15461726. doi: 10.1038/s41593-021-00917-2.

R. Desimone, T. D. Albright, C. G. Gross, and C. Bruce. Stimulus-selective properties of inferior temporal neurons in the macaque. *Journal of Neuroscience*, 4(8), 1984. ISSN 02706474. doi: 10.1523/jneurosci.04-08-02051.1984.

K. Dhamdhere, Q. Yan, and M. Sundararajan. How important is a neuron? In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

V. Di Lollo. The feature-binding problem is an ill-posed problem, 2012. ISSN 13646613.

R. Diamond and S. Carey. Why Faces Are and Are Not Special. An Effect of Expertise. *Journal of Experimental Psychology: General*, 115(2), 1986. ISSN 00963445. doi: 10.1037/0096-3445.115.2.107.

J. J. DiCarlo and D. D. Cox. Untangling invariant object recognition. *Trends in Cognitive Sciences*, 2007. ISSN 13646613. doi: 10.1016/j.tics.2007.06.010.

J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object

recognition? *Neuron*, 73(3), 2012. ISSN 08966273. doi: 10.1016/j.neuron.2012.01.010.

M. Diesmann, M. O. Gewaltig, and A. Aertsen. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 1999. doi: 10.1038/990101.

Y. Dong, S. Mihalas, F. Qiu, R. von der Heydt, and E. Niebur. Synchrony and the binding problem in macaque visual cortex. *Journal of Vision*, 2008. ISSN 1534-7362. doi: 10.1167/8.7.30.

Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting Adversarial Attacks with Momentum. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00957.

M. Donk. Illusory conjunctions are an illusion: The effects of target-nontarget similarity on conjunction and feature errors. *Journal of Experimental Psychology: Human Perception and Performance*, 25(5), 1999. ISSN 00961523. doi: 10.1037/0096-1523.25.5.1207.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*, 2021.

M. Dujmović, G. Malhotra, and J. S. Bowers. What do adversarial images tell us about human vision? *eLife*, 9, 2020. ISSN 2050084X. doi: 10.7554/ELIFE.55978.

J. Duncan and G. W. Humphreys. Visual Search and Stimulus Similarity. *Psychological Review*, 96(3), 1989. ISSN 0033295X. doi: 10.1037/0033-295X.96.3.433.

R. Eckhorn, R. Bauer, W. Jordan, M. Brosch, W. Kruse, M. Munk, and H. J. Reitboeck. Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, 60(2), 1988. ISSN 0340-1200. doi: 10.1007/bf00202899.

R. Egger, Y. Tupikov, M. Elmaleh, K. A. Katlowitz, S. E. Benezra, M. A. Picardo, F. Moll, J. Kornfeld, D. Z. Jin, and M. A. Long. Local Axonal Conduction Shapes the Spa-

tiotemporal Properties of Neural Sequences. *Cell*, 183(2), 2020. ISSN 10974172. doi: 10.1016/j.cell.2020.09.019.

A. Eguchi and S. M. Stringer. Neural network model develops border ownership representation through visually guided learning. *Neurobiology of Learning and Memory*, 136:147–165, 2016. ISSN 10959564. doi: 10.1016/j.nlm.2016.10.007. URL http://dx.doi.org/10.1016/j.nlm.2016.10.007.

A. Eguchi, B. Mender, B. Evans, G. Humphreys, and S. Stringer. Computational modeling of the neural representation of object shape in the primate ventral visual system. *Front. Comput. Neurosci.*, 9(August), 2015. doi: 10.3389/fncom.2015.00100. URL http://journal.frontiersin.org/Article/10.3389/fncom.2015.00100/abstract.

A. Eguchi, J. B. Isbister, N. Ahmad, and S. Stringer. The emergence of polychronization and feature binding in a spiking neural network model of the primate ventral visual system. *Psychological Review*, 2018. ISSN 0033295X. doi: 10.1037/rev0000103.

B. Ehret, C. Henning, M. Cervera, A. Meulemans, J. von Oswald, and B. Grewe. Continual Learning in Recurrent Neural Networks. *International Conference on Learning Representations*, 2021.

H. Eichenbaum. Memory binding in hippocampal relational networks. In *Handbook of Binding and Memory: Perspectives from Cognitive Neuroscience*. 2012. doi: 10.1093/acprof:oso/9780198529675.003.0002.

H. Eichenbaum and N. J. Cohen. Can We Reconcile the Declarative Memory and Spatial Navigation Views on Hippocampal Function?, 2014. ISSN 10974199.

M. C. Elliffe, E. T. Rolls, and S. M. Stringer. Invariant recognition of feature combinations in the visual system. *Biological Cybernetics*, 2002. ISSN 03401200. doi: 10.1007/s004220100284.

G. F. Elsayed, N. Papernot, S. Shankar, A. Kurakin, B. Cheung, I. Goodfellow, and J. Sohl-Dickstein. Adversarial examples that fool both computer vision and time-

limited humans. In *Advances in Neural Information Processing Systems*, volume 2018-December, 2018.

B. Evans, G. Malhotra, and J. Bowers. Biological convolutions improve DNN robustness to noise and generalisation. *Neural Networks : 2021 Special Issue on AI and Brain Science*, 2022.

B. D. Evans and S. M. Stringer. Transformation-invariant visual representations in self-organizing spiking neural networks. *Frontiers in Computational Neuroscience*, 6(JUL), 2012. ISSN 16625188. doi: 10.3389/fncom.2012.00046.

L. Fei-Fei, R. VanRullen, C. Koch, and P. Perona. Why does natural scene categorization require little attention? Exploring attentional requirements for natural and synthetic stimuli. *Visual Cognition*, 12(6), 2005. ISSN 13506285. doi: 10.1080/13506280444000571.

D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1), 1991. ISSN 10473211. doi: 10.1093/cercor/1.1.1.

I. R. Fiete, Y. Burak, and T. Brookings. What grid cells convey about rat location. *Journal of Neuroscience*, 28(27), 2008. ISSN 02706474. doi: 10.1523/JNEUROSCI.5684-07.2008.

E. Fix and J. L. Hodges. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3), 1989. ISSN 03067734. doi: 10.2307/1403797.

T. Flesch, J. Balaguer, R. Dekker, H. Nili, and C. Summerfield. Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences of the United States of America*, 115(44), 2018. ISSN 10916490. doi: 10.1073/pnas.1800755115.

N. Ford, J. Gilmer, N. Carlini, and E. D. Cubuk. Adversarial examples are a natural consequence of test error in noise. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, 2019.

T. Foulsham and A. Kingstone. Fixation-dependent memory for natural scenes: An experimental test of scanpath theory. *Journal of Experimental Psychology: General*, 142 (1), 2013a. ISSN 00963445. doi: 10.1037/a0028227.

T. Foulsham and A. Kingstone. Optimal and preferred eye landing positions in objects and scenes. *Quarterly Journal of Experimental Psychology*, 66(9), 2013b. ISSN 17470218. doi: 10.1080/17470218.2012.762798.

S. R. Friedman-Hill, L. C. Robertson, and A. Treisman. Parietal contributions to visual feature binding: Evidence from a patient with bilateral lesions. *Science*, 269(5225), 1995. ISSN 00368075. doi: 10.1126/science.7638604.

Y. Fu, S. Zhang, S. Wu, C. Wang, and Y. Lin. Patch-Fool : Are Vision Transformers Always Robust Against Adversarial Perturbations? *International Conference on Learning Representations 2022*, 2022.

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 1980. ISSN 03401200. doi: 10.1007/BF00344251.

L. Gabernet, S. P. Jadhav, D. E. Feldman, M. Carandini, and M. Scanziani. Somatosensory integration controlled by dynamic thalamocortical feed-forward inhibition. *Neuron*, 48 (2), 2005. ISSN 08966273. doi: 10.1016/j.neuron.2005.09.022.

A. K. Garg, P. Li, M. S. Rashid, and E. M. Callaway. Color and orientation are jointly coded and spatially organized in primate primary visual cortex. *Science*, 364(6447), 2019. ISSN 10959203. doi: 10.1126/science.aaw5868.

I. Gauthier and M. J. Tarr. Becoming a 'Greeble' expert: Exploring mechanisms for face recognition. *Vision Research*, 37(12), 1997. ISSN 00426989. doi: 10.1016/S0042-6989(96)00286-6.

I. Gauthier, P. Skudlarski, J. C. Gore, and A. W. Anderson. Expertise for cars and birds recruits brain areas involved in face recognition. *Nature Neuroscience*, 3(2), 2000. ISSN 10976256. doi: 10.1038/72140.

R. Geirhos, H. H. Schütt, C. R. Medina Temme, M. Bethge, J. Rauber, and F. A. Wich-

mann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, volume 2018-December, 2018.

R. Geirhos, C. Michaelis, F. A. Wichmann, P. Rubisch, M. Bethge, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

R. Geirhos, K. Narayanappa, B. Mitzkus, T. Thieringer, M. Bethge, F. Wichmann, and W. Brendel. Partial success in closing the gap between human and machine vision. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, B. Marthi, X. Lou, Z. Meng, Y. Liu, H. Wang, A. Lavin, and D. S. Phoenix. A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science*, 358 (6368), 2017. ISSN 10959203. doi: 10.1126/science.aag2612.

W. Gerstner and W. Kistler. *Spiking Neuron Models : Single Neurons, Populations, Plasticity* . Cambridge University Press, 2002.

W. Gerstner, R. Kempter, J. L. Van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595), 1996. ISSN 00280836. doi: 10.1038/383076a0.

G. M. Ghose and J. Maunsell. Specialized representations in visual cortex: A role for binding? *Neuron*, 24(1), 1999. ISSN 08966273. doi: 10.1016/S0896-6273(00)80823-5.

E. M. Gibson, D. Purger, C. W. Mount, A. K. Goldstein, G. L. Lin, L. S. Wood, I. Inema, S. E. Miller, G. Bieri, J. B. Zuchero, B. A. Barres, P. J. Woo, H. Vogel, and M. Monje. Neuronal activity promotes oligodendrogenesis and adaptive myelination in the mammalian brain. *Science*, 344(6183), 2014. ISSN 10959203. doi: 10.1126/science.1252304.

C. D. Gilbert. Laminar differences in receptive field properties of cells in cat primary

visual cortex. *The Journal of Physiology*, 268(2), 1977. ISSN 14697793. doi: 10.1113/jphysiol.1977.sp011863.

J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, I. Goodfellow, and G. Brain. The Relationship Between High-Dimensional Geometry and Adversarial Examples. *arXiv:1801.00634*, 2018.

A. Goldstein, Z. Zada, E. Buchnik, M. Schain, A. Price, B. Aubrey, S. Nastase, A. Feder, D. Emanuel, A. Cohen, A. Jansen, H. Gazula, G. Choe, A. Rao, C. Kim, C. Casto, L. Fanda, W. Doyle, D. Friedman, P. Dugan, L. Melloni, R. Reichart, S. Devore, A. Flinker, L. Hasenfratz, O. Levy, A. Hassidim, M. Brenner, Y. Matias, K. Norman, O. Devinsky, and U. Hasson. Shared computational principles for language processing in humans and deep language models. *Nature Neuroscience*, 25:369–380, 5 2022.

I. Goodfellow, J. Schlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*, 2015.

I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning, 2016.

S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv*, 2020. ISSN 23318422.

S. Gowal, P.-S. Huang, A. van den Oord, T. Mann, and P. Kohli. Self-Supervised Adversarial Robustness for the Low-Label, High-Data Regime. *International Conference on Learning Representations*, 2021.

M. Graf. Coordinate transformations in object recognition. *Psychological Bulletin*, 132 (6), 2006. ISSN 00332909. doi: 10.1037/0033-2909.132.6.920.

A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626), 2016. ISSN 14764687. doi: 10.1038/nature20101.

C. M. Gray. The temporal correlation hypothsis of visual feature integration: Still alive and well. *Neuron*, 24:31–47, 1999.

C. M. Gray and W. Singer. Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 86(5), 1989. ISSN 00278424. doi: 10.1073/pnas.86.5.1698.

K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, 2019.

K. Greff, S. van Steenkiste, and J. Schmidhuber. On the Binding Problem in Artificial Neural Networks. *arXiv*, arXiv:2012.05208, 2020.

J. Grimes. On the Failure to Detect Changes in Scenes across Saccades. *Perception*, 1996. doi: 10.1093/acprof.

L. N. Groschner, L. Chan Wah Hak, R. Bogacz, S. DasGupta, and G. Miesenböck. Dendritic Integration of Sensory Evidence in Perceptual Decision-Making. *Cell*, 173(4), 2018. ISSN 10974172. doi: 10.1016/j.cell.2018.03.075.

S. Grossberg. Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, 37, 2013. ISSN 08936080. doi: 10.1016/j.neunet.2012.09.017.

M. Guise, A. Knott, and L. Benuskova. A Bayesian model of polychronicity. *Neural computation*, 26:2052–2073, 2014. ISSN 1530888X. doi: 10.1162/NECO{_}a{_}00620.

S. Gupta, M. Zhang, C.-C. Wu, J. Wolfe, and G. Kreiman. Visual Search Asymmetry: Deep Nets and Humans Share Similar Inherent Biases. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

Z. M. Hafed and C. Y. Chen. Sharper, Stronger, Faster Upper Visual Field Representation in Primate Superior Colliculus. *Current Biology*, 26(13), 2016. ISSN 09609822. doi: 10.1016/j.cub.2016.04.059.

T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 2005. ISSN 00280836. doi: 10.1038/nature03721.

D. J. Hagler. Visual field asymmetries in visual evoked responses. *Journal of Vision*, 14 (14), 2014. ISSN 15347362. doi: 10.1167/14.14.13.

Y. Han, G. Roig, G. Geiger, and T. Poggio. Scale and translation-invariance for novel objects in human vision. *Scientific Reports*, 10(1), 2020. ISSN 20452322. doi: 10.1038/s41598-019-57261-6.

K. D. Harris, D. A. Henze, H. Hirase, X. Leinekugel, G. Dragoi, A. Czurkó, and G. Buzsáki. Spike train dynamics predicts theta-related phase precession in hippocampal pyramidal cells. *Nature*, 417(6890), 2002. ISSN 00280836. doi: 10.1038/nature00808.

D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-Inspired Artificial Intelligence. *Neuron*, 95(2):245–258, 2017. ISSN 10974199. doi: 10.1016/j.neuron.2017.06.011. URL http://dx.doi.org/10.1016/j.neuron.2017.06.011.

M. Häusser and A. Roth. Estimating the Time Course of the Excitatory Synaptic Conductance in Neocortical Pyramidal Cells Using a Novel Voltage Jump Method. *The Journal of Neuroscience*, 1997. ISSN 0270-6474. doi: 10.1523/jneurosci.17-20-07606.1997.

M. N. Havenith, S. Yu, J. Biederlack, N.-H. Chen, W. Singer, and D. Nikolic. Synchrony Makes Neurons Fire in Sequence, and Stimulus Properties Determine Who Is Ahead. *Journal of Neuroscience*, 2011. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2817-10.2011.

J. Hawkins and S. Ahmad. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Frontiers in neural circuits*, 10, 2016. ISSN 16625110. doi: 10.3389/fncir.2016.00023.

J. Hawkins, S. Ahmad, and Y. Cui. A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Frontiers in Neural Circuits*, 11(October):1–18,

2017. ISSN 1662-5110. doi: 10.3389/fncir.2017.00081. URL http://journal.frontiersin.org/article/10.3389/fncir.2017.00081/full.

J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, 2019. ISSN 16625110. doi: 10.3389/fncir.2018.00121.

M. Hayhoe, J. Lachter, and J. Feldman. Integration of form across saccadic eye movements. *Perception*, 20(3), 1991. ISSN 03010066. doi: 10.1068/p200393.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 770–778. IEEE Computer Society, 12 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90.

S. Hedayati, R. O'Donnell, and B. Wyble. A model of working memory for latent representations. *Nature Human Behaviour*, 2022.

D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, 9(2), 1992. ISSN 14698714. doi: 10.1017/S0952523800009640.

M. Hein, M. Andriushchenko, and J. Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019. doi: 10.1109/CVPR.2019.00013.

O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. v. d. Oord. Data-Efficient Image Recognition with Contrastive Predictive Coding. *Proceedings of the 37th International Conference on Machine Learning*, 5 2020. URL http://arxiv.org/abs/1905.09272.

E. Higgins and K. Rayner. Transsaccadic processing: stability, integration, and the potential role of remapping. *Attention, Perception, and Psychophysics*, 77(1), 2014. ISSN 1943393X. doi: 10.3758/s13414-014-0751-y.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. B-VAE: Learning basic visual concepts with a constrained variational

framework. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

I. Higgins, L. Chang, V. Langston, D. Hassabis, C. Summerfield, D. Tsao, and M. Botvinick. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nature Communications*, 12(1), 2021. ISSN 20411723. doi: 10.1038/s41467-021-26751-5.

J. H. Hines, A. M. Ravanelli, R. Schwindt, E. K. Scott, and B. Appel. Neuronal activity biases axon selection for myelination in vivo. *Nature Neuroscience*, 18(5), 2015. ISSN 15461726. doi: 10.1038/nn.3992.

G. E. Hinton, J. L. Mcclelland, and D. E. Rumelhart. Distributed representations (memory storage). In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, 1986.

K. Hoang, A. Pitti, J. F. Goudou, J. Y. Dufour, and P. Gaussier. Active vision: On the relevance of a bio-inspired approach for object detection. *Bioinspiration and Biomimetics*, 15(2), 2020. ISSN 17483190. doi: 10.1088/1748-3190/ab504c.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9 (8), 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.

S. Hochstein and M. Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36(5), 2002. ISSN 08966273. doi: 10.1016/S0896-6273(02) 01091-7.

L. Holm, J. Eriksson, and L. Andersson. Looking as if you know: Systematic object inspection precedes object recognition. *Journal of Vision*, 8(4), 2008. ISSN 15347362. doi: 10.1167/8.4.14.

B. Hommel and L. S. Colzato. When an object is more than a binding of its features: Evidence for two mechanisms of visual feature integration. *Visual Cognition*, 17(1-2), 2009. ISSN 13506285. doi: 10.1080/13506280802349787.

H. Hong, D. L. Yamins, N. J. Majaj, and J. J. Dicarlo. Explicit information for category-

orthogonal object properties increases along the ventral stream. *Nature Neuroscience*, 19(4), 2016. ISSN 15461726. doi: 10.1038/nn.4247.

Y. Hong, L. Yi, J. Tenenbaum, A. Torralba, and C. Gan. PTR: A Benchmark for Part-based Conceptual, Relational, and Physical Reasoning. *35th Conference on Neural Information Processing Systems (NeurIPS 2021), virtual.*, 2021.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8), 1982. ISSN 00278424. doi: 10.1073/pnas.79.8.2554.

J. J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535), 1995. ISSN 00280836. doi: 10.1038/376033a0.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.243.

Y. Huang, J. Gornet, S. Dai, Z. Yu, T. Nguyen, D. Tsao, and A. Anandkumar. Neural Networks with Recurrent Generative Feedback. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

D. Hubel and T. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3), 1959. ISSN 14697793. doi: 10.1113/jphysiol.1959.sp006308.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1), 1962. ISSN 14697793. doi: 10.1113/jphysiol.1962.sp006837.

J. E. Hummel and I. Biederman. Dynamic Binding in a Neural Network for Shape Recognition. *Psychological Review*, 99(3):480–517, 1992. ISSN 0033295X. doi: 10.1037/0033-295X.99.3.480.

G. W. Humphreys. Feature confirmation in object perception: Feature integration theory

26 years on from the Treisman Bartlett lecture. *Quarterly Journal of Experimental Psychology*, 69(10), 2016. ISSN 17470226. doi: 10.1080/17470218.2014.988736.

G. W. Humphreys, J. Hodsoll, and M. J. Riddoch. Fractionating the Binding Process: Neuropsychological Evidence From Reversed Search Efficiencies. *Journal of Experimental Psychology: Human Perception and Performance*, 35(3), 2009. ISSN 00961523. doi: 10.1037/a0013705.

P. R. Huttenlocher and A. S. Dabholkar. Regional differences in synaptogenesis in human cerebral cortex. *Journal of Comparative Neurology*, 387(2), 1997. ISSN 00219967. doi: 10.1002/(SICI)1096-9861(19971020)387:2<167::AID-CNE1>3.0.CO;2-Z.

T. Hwu, A. Y. Wang, N. Oros, and J. L. Krichmar. Adaptive robot path planning using a spiking neuron algorithm with axonal delays. *IEEE Transactions on Cognitive and Developmental Systems*, 10(2), 2018. ISSN 23798939. doi: 10.1109/TCDS.2017.2655539.

A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial Examples are not Bugs, they are Features. In *Neural Information Processing Systems (NIPS)*, 2019.

J. Isbister and S. Stringer. Conference Submitted Title : "The emergence of spatiotemporal spike patterns and feature binding relations within a spiking neural network model of the primate visual cortex: a cortical implementation of capsule networks"; Final Title "A Generalisation Theory of Spike-Time Relativity : Precise spike timing as a cortical implementation of capsule networks?". In *CNS*, page 116, Seattle, 2018.

J. B. Isbister, A. Eguchi, N. Ahmad, J. Galeazzi, M. Buckley, and S. Stringer. A new approach to solving the feature binding problem in primate vision. *Interface Focus*, 8:–, 2018.

E. M. Izhikevich. Polychronization: Computation with Spikes. *Neural Computation*, 18(2):245–282, 2006. ISSN 0899-7667. doi: 10.1162/089976606775093882. URL http://www.mitpressjournals.org/doi/10.1162/089976606775093882.

G. Jacob, R. T. Pramod, H. Katti, and S. P. Arun. Qualitative similarities and differences in visual object representations between brains and deep networks. *Nature Communications*, 12(1), 2021. ISSN 20411723. doi: 10.1038/s41467-021-22078-3.

J. H. Jacobsen, A. Smeulders, and E. Oyallon. I-RevNet: Deep invertible networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

M. Jadi, A. Polsky, J. Schiller, and B. W. Mel. Location-dependent effects of inhibition on local spiking in pyramidal neuron dendrites. *PLoS Computational Biology*, 8(6), 2012. ISSN 1553734X. doi: 10.1371/journal.pcbi.1002550.

A. Jalal, A. Ilyas, C. Daskalakis, and A. G. Dimakis. The Robust Manifold Defense: Adversarial Training using Generative Models. *arXiv preprint arXiv:1712.09196*, 12 2017. URL http://arxiv.org/abs/1712.09196.

U. Jang, S. Jah, and S. Jah. On the Need for Topology-Aware Generative Models for Manifold-Based Defenses. *International Conference on Learning Representations*, 2020.

L. A. Jeffress. A place theory of sound localization. *Journal of Comparative and Physiological Psychology*, 41(1), 1948. ISSN 00219940. doi: 10.1037/h0061495.

L. C. Johansson and P. Henningsson. Butterflies fly using efficient propulsive clap mechanism owing to flexible wings. *Journal of the Royal Society Interface*, 18(174), 2021. ISSN 17425662. doi: 10.1098/rsif.2020.0854.

J. B. Julian, A. T. Keinath, G. Frazzetta, and R. A. Epstein. Human entorhinal cortex represents visual space using a boundary-anchored grid. *Nature Neuroscience*, 21(2), 2018. ISSN 15461726. doi: 10.1038/s41593-017-0049-1.

S. Kamath, A. Deshpande, K. Subrahmanyam, and V. Balasubramanian. Can we have it all? On the Trade-off between Spatial and Adversarial Robustness of Neural Networks. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

P. Kanerva. Sparse distributed memory and related models. In *Associative Neural Memories: Theory and Implementation*. 1993.

J. Kanev, A. Koutsou, C. Christodoulou, and K. Obermayer. Integrator or coincidence detector: A novel measure based on the discrete reverse correlation to determine a Neuron's operational mode. *Neural Computation*, 28(10), 2016. ISSN 1530888X. doi: 10.1162/NECO{_}a{_}00875.

I. Kang and J. H. Maunsell. The correlation of neuronal signals with behavior at different levels of visual cortex and their relative reliability for behavioral decisions. *Journal of Neuroscience*, 40(19), 2020. ISSN 15292401. doi: 10.1523/JNEUROSCI.2587-19.2020.

K. Kar, J. Kubilius, K. Schmidt, E. B. Issa, and J. J. DiCarlo. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature Neuroscience*, 22(6), 2019. ISSN 15461726. doi: 10.1038/s41593-019-0392-5.

P. Kara, P. Reinagel, and R. C. Reid. Low response variability in simultaneously recorded retinal, thalamic, and cortical neurons. *Neuron*, 27(3), 2000. ISSN 08966273. doi: 10.1016/S0896-6273(00)00072-6.

O. S. Kayhan and J. C. van Gemert. On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. doi: 10.1109/CVPR42600.2020.01428.

R. Kempter, W. Gerstner, and J. L. Van Hemmen. How the threshold of a neuron determines its capacity for coincidence detection. In *BioSystems*, volume 48, 1998. doi: 10.1016/S0303-2647(98)00055-0.

S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99, 2018. ISSN 18792782. doi: 10.1016/j.neunet.2017.12.005.

M. Khoury and D. Hadfield-Menell. On the Geometry of Adversarial Examples. *arXiv preprint arXiv:1811.00525*, 11 2018. URL http://arxiv.org/abs/1811.00525.

N. J. Killian, M. J. Jutras, and E. A. Buffalo. A map of visual space in the primate entorhinal cortex. *Nature*, 491(7426), 2012. ISSN 00280836. doi: 10.1038/nature11587.

S. H. Kim and J. Feldman. Globally inconsistent figure/ground relations induced by a negative part. *Journal of Vision*, 9(10), 2009. ISSN 15347362. doi: 10.1167/9.10.8.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13), 2017. ISSN 10916490. doi: 10.1073/pnas.1611835114.

M. Klukas, M. Lewis, and I. Fiete. Efficient and flexible representation of higher-dimensional cognitive variables with grid cells. *PLoS Computational Biology*, 16(4), 2020. ISSN 15537358. doi: 10.1371/journal.pcbi.1007796.

C. Koch, M. Rapp, and I. Segev. A brief history of time (constants). *Cerebral Cortex*, 6 (2), 1996. ISSN 10473211. doi: 10.1093/cercor/6.2.93.

P. König, A. K. Engel, and W. Singer. Integrator or coincidence detector? The role of the cortical neuron revisited. *Trends in Neurosciences*, 19(4), 1996. ISSN 01662236. doi: 10.1016/S0166-2236(96)80019-1.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *Science Department, University of Toronto, Tech.*, 2009. ISSN 1098-6596. doi: 10.1.1.222.9220.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*, 2012. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

G. Króliczak, M. A. Goodale, and G. K. Humphrey. The effects of different aperture-viewing conditions on the recognition of novel objects. *Perception*, 32(10), 2003. ISSN 03010066. doi: 10.1068/p3443.

J. Kubilius, M. Schrimpf, K. Kar, R. Rajalingham, H. Hong, N. J. Majaj, E. B. Issa, P. Bashivan, J. Prescott-Roy, K. Schmidt, A. Nayebi, D. Bear, D. L. Yamins, and J. J. DiCarlo. Brain-like object recognition with high-performing shallow recurrent ANNs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

R. D. Kumbhani, M. J. Nolt, and L. A. Palmer. Precision, Reliability, and Information-Theoretic Analysis of Visual Thalamocortical Neurons. *Journal of Neurophysiology*, 2007. ISSN 0022-3077. doi: 10.1152/jn.00900.2006.

A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, 2019.

N. Kuwabara and N. Suga. Delay lines and amplitude selectivity are created in subthalamic auditory nuclei: The brachium of the inferior colliculus of the mustached bat. *Journal of Neurophysiology*, 69(5), 1993. ISSN 00223077. doi: 10.1152/jn.1993.69.5.1713.

C. Laidlaw, S. Singla, and S. Feizi. Perceptual Adversarial Robustness: Defense Against Unseen Threat Models. *International Conference on Learning Representations*, 2021. ISSN 23318422.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266), 2015. ISSN 10959203. doi: 10.1126/science.aab3050.

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, (2012):1–101, 2016. ISSN 14691825. doi: 10.1017/S0140525X16001837.

H. Larochelle and G. Hinton. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.

M. Lavidor and V. Walsh. The nature of foveal representation. *Nature Reviews Neuroscience*, 5(9), 2004. ISSN 1471003X. doi: 10.1038/nrn1498.

N. Leadholm, M. Lewis, and S. Ahmad. Grid Cell Path Integration For Movement-

Based Visual Object Recognition. *The 32nd British Machine Vision Conference (BMVC 2021)*, 2021.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, number November, pages 1–46, 1998. ISBN 0018-9219. doi: 10.1109/5.726791.

Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

H. Lee, H. Bae, and S. Yoon. Gradient Masking of Label Smoothing in Adversarial Robustness. *IEEE Access*, 9, 2021. ISSN 21693536. doi: 10.1109/ACCESS.2020.3048120.

S. Lefort, C. Tomm, J. C. Floyd Sarria, and C. C. Petersen. The Excitatory Neuronal Network of the C2 Barrel Column in Mouse Primary Somatosensory Cortex. *Neuron*, 61(2), 2009. ISSN 08966273. doi: 10.1016/j.neuron.2008.12.020.

J. F. Léger, E. A. Stern, A. Aertsen, and D. Heck. Synaptic integration in rat frontal cortex shaped by network activity. *Journal of Neurophysiology*, 93(1), 2005. ISSN 00223077. doi: 10.1152/jn.00067.2003.

S. R. Lehky, M. E. Sereno, and A. B. Sereno. Characteristics of eye-position gain field populations determine geometry of visual space. *Frontiers in Integrative Neuroscience*, 9(JAN2016), 2016. ISSN 16625145. doi: 10.3389/fnint.2015.00072.

M. Lewis, S. Purdy, S. Ahmad, and J. Hawkins. Locations in the neocortex: A theory of sensorimotor object recognition using cortical grid cells. *Frontiers in Neural Circuits*, 2019. ISSN 16625110. doi: 10.3389/fncir.2019.00022.

F. F. Li, R. VanRullen, C. Koch, and P. Perona. Rapid natural scene categorization in the near absence of attention. *Proceedings of the National Academy of Sciences of the United States of America*, 99(14), 2002. ISSN 00278424. doi: 10.1073/pnas.092277599.

Z. Li, W. Brendel, E. Y. Walker, E. Cobos, T. Muhammad, J. Reimer, M. Bethge, F. H. Sinz, X. Pitkow, and A. S. Tolias. Learning from brains how to regularize machines. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-Centric Learning with Slot Attention. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 6 2020. URL http://arxiv.org/abs/2006.15055.

N. K. Logothetis, J. Pauls, and T. Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5), 1995. ISSN 09609822. doi: 10.1016/S0960-9822(95)00108-4.

M. London, A. Roth, L. Beeren, M. Häusser, and P. E. Latham. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466(7302), 2010. ISSN 00280836. doi: 10.1038/nature09086.

X. Long and S. J. Zhang. A novel somatosensory spatial navigation system outside the hippocampal formation. *Cell Research*, 2021. ISSN 17487838. doi: 10.1038/s41422-020-00448-8.

D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

Y. Lu, J. Yin, Z. Chen, H. Gong, Y. Liu, L. Qian, X. Li, R. Liu, I. M. Andolina, and W. Wang. Revealing Detail along the Visual Hierarchy: Neural Clustering Preserves Acuity from V1 to V4. *Neuron*, 98(2), 2018. ISSN 10974199. doi: 10.1016/j.neuron.2018.03.009.

S. J. Luck and E. K. Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390(6657), 1997. ISSN 00280836. doi: 10.1038/36846.

L. Lücken, D. P. Rosin, V. M. Worlitzer, and S. Yanchuk. Pattern reverberation in networks of excitable systems with connection delays. *Chaos*, 27(1), 2017. ISSN 10541500. doi: 10.1063/1.4971971.

A. Luczak, B. L. McNaughton, and K. D. Harris. Packet-based communication in the cortex. *Nature Reviews Neuroscience*, 16(12), 2015. ISSN 14710048. doi: 10.1038/nrn4026.

C. J. MacDowell and T. J. Buschman. Low-Dimensional Spatiotemporal Dynamics Un-

derlie Cortex-wide Neural Activity. *Current Biology*, 30(14), 2020. ISSN 18790445. doi: 10.1016/j.cub.2020.04.090.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

R. Malach, J. B. Reppas, R. R. Benson, K. K. Kwong, H. Jiang, W. A. Kennedy, P. J. Ledden, T. J. Brady, B. R. Rosen, and R. B. Tootell. Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 92(18), 1995. ISSN 00278424. doi: 10.1073/pnas.92.18.8135.

G. Malhotra, B. D. Evans, and J. S. Bowers. Hiding a plane with a pixel: examining shape-bias in CNNs and the benefit of building in biological constraints. *Vision Research*, 174, 2020. ISSN 18785646. doi: 10.1016/j.visres.2020.04.013.

S. G. Manohar, N. Zokaei, S. J. Fallon, T. P. Vogels, and M. Husain. Neural mechanisms of attending to items in working memory. *Neuroscience and Biobehavioral Reviews*, 101, 2019. ISSN 18737528. doi: 10.1016/j.neubiorev.2019.03.017.

H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 1997. ISSN 00368075. doi: 10.1126/science.275.5297.213.

D. Marr. Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information. *MIT Press*, 1982.

T. Masquelier. STDP Allows Close-to-Optimal Spatiotemporal Spike Pattern Detection by Single Coincidence Detector Neurons. *Neuroscience*, 389, 2018. ISSN 18737544. doi: 10.1016/j.neuroscience.2017.06.032.

T. Masquelier and S. J. Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, 3(2), 2007. ISSN 1553734X. doi: 10.1371/journal.pcbi.0030031.

T. Masquelier, R. Guyonneau, and S. J. Thorpe. Spike timing dependent plasticity finds

the start of repeating patterns in continuous spike trains. *PLoS ONE*, 3(1), 2008. ISSN 19326203. doi: 10.1371/journal.pone.0001377.

T. Masquelier, R. Guyonneau, and S. J. Thorpe. Competitive STDP-based spike pattern learning. *Neural Computation*, 21(5), 2009. ISSN 1530888X. doi: 10.1162/neco.2008.06-08-804.

J. L. McClelland, D. E. Rumelhart, and PDP Research Group. *Parallel Distributed Processing: Psychological and Biological Models (Vol. 2)*. MIT Press, 1986.

M. McCloskey and N. J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C), 1989. ISSN 00797421. doi: 10.1016/S0079-7421(08)60536-8.

G. W. McConkie and K. Rayner. Identifying the span of the effective stimulus in reading: Literature review and theories of reading. In *Theoretical models and processes of reading*. 1976.

I. A. McKenzie, D. Ohayon, H. Li, J. P. De Faria, B. Emery, K. Tohyama, and W. D. Richardson. Motor skill learning requires active central myelination. *Science*, 346 (6207), 2014. ISSN 10959203. doi: 10.1126/science.1254960.

M. R. Mehta, A. K. Lee, and M. A. Wilson. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature*, 417(6890), 2002. ISSN 00280836. doi: 10.1038/nature00807.

B. M. Mender and S. M. Stringer. A model of self-organizing head-centered visual responses in primate parietal areas. *PLoS ONE*, 8(12), 2013. ISSN 19326203. doi: 10.1371/journal.pone.0081406.

P. M. Milner. A model for visual shape recognition. *Psychological Review*, 1974. ISSN 0033295X. doi: 10.1037/h0037149.

V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, volume 3, 2014.

M. L. Montero, C. J. H. Ludwig, R. Ponte Costa, G. Malhotra, and J. S. Bowers. The Role of Disentanglement in Generalisation. *ICLR*, 2021.

S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.282.

E. I. Moser, E. Kropff, and M. B. Moser. Place cells, grid cells, and the brain's spatial representation system. *Annual Review of Neuroscience*, 31, 2008. ISSN 0147006X. doi: 10.1146/annurev.neuro.31.061307.090723.

V. B. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4), 1997. ISSN 00068950. doi: 10.1093/brain/120.4.701.

N. Mu and J. Gilmer. MNIST-C: A Robustness Benchmark for Computer Vision. *ICML 2019 Workshop on Uncertainty and Ro- bustness in Deep Learning*, 2019.

D. Mumford. The Convergence of AI code and Cortical Functioning - a Commentary. *arXiv*, arXiv:2010.09101, 2020.

E. A. Murray, T. J. Bussey, and L. M. Saksida. Visual perception and memory: A new view of medial temporal lobe function in primates and rodents. *Annual Review of Neuroscience*, 30, 2007. ISSN 0147006X. doi: 10.1146/annurev.neuro.29.051605.113046.

V. Nair and G. E. Hinton. Rectified linear units improve Restricted Boltzmann machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010. ISBN 9781605589077.

J. Nassar, P. Sokol, S. Chung, K. Harris, and I. Park. On 1/n neural representation and robustness. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

M. Nau, T. Navarro Schröder, J. L. Bellmund, and C. F. Doeller. Hexadirectional coding of visual space in human entorhinal cortex. *Nature Neuroscience*, 21(2), 2018. ISSN 15461726. doi: 10.1038/s41593-017-0050-8.

I. Nauhaus, L. Busse, M. Carandini, and D. L. Ringach. Stimulus contrast modulates functional connectivity in visual cortex. *Nature Neuroscience*, 12(1), 2009. ISSN 10976256. doi: 10.1038/nn.2232.

D. M. Navarro, H. E. Smithson, and S. M. Stringer. A Modeling Study of the Emergence of Eye Position Gain Fields Modulating the Responses of Visual Neurons in the Brain. *Frontiers in Neural Circuits*, 14, 2020. ISSN 16625110. doi: 10.3389/fncir.2020.00030.

A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, 2004. ISBN 1581138385. doi: 10.1145/1015330.1015435.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, 2015. doi: 10.1109/CVPR.2015.7298640.

A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, 2016.

Q. Y. Nie, H. J. Müller, and M. Conci. Hierarchical organization in visual working memory: From global ensemble to individual object structure. *Cognition*, 159, 2017. ISSN 18737838. doi: 10.1016/j.cognition.2016.11.009.

D. Noton and L. Stark. Scanpaths in eye movements during pattern perception. *Science*, 171(3968), 1971. ISSN 00368075. doi: 10.1126/science.171.3968.308.

J. O'Keefe and D. H. Conway. Hippocampal place units in the freely moving rat: Why they fire where they fire. *Experimental Brain Research*, 31(4), 1978. ISSN 00144819. doi: 10.1007/BF00239813.

J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1), 1971. ISSN 00068993. doi: 10.1016/0006-8993(71)90358-1.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 1996. ISSN 00280836. doi: 10.1038/381607a0.

B. A. Olshausen, C. H. Anderson, and D. C. Van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11), 1993. ISSN 02706474. doi: 10.1523/jneurosci.13-11-04700.1993.

E. B. O'Neil, A. D. Cate, and S. Köhler. Perirhinal cortex contributes to accuracy in recognition memory and perceptual discriminations. *Journal of Neuroscience*, 29(26), 2009. ISSN 02706474. doi: 10.1523/JNEUROSCI.0374-09.2009.

R. C. O'Reilly and J. W. Rudy. Conjunctive representations in learning and memory: Principles of cortical and hippocampal function. *Psychological Review*, 108(2), 2001. ISSN 0033295X. doi: 10.1037/0033-295X.108.2.311.

T. Orlov and E. Zohary. Object representations in human visual cortex formed through temporal integration of dynamic partial shape views. *Journal of Neuroscience*, 38(3), 2018. ISSN 15292401. doi: 10.1523/JNEUROSCI.1318-17.2017.

S. Pajevic, P. J. Basser, and R. D. Fields. Role of myelin plasticity in oscillations and synchrony of neuronal activity. *Neuroscience*, 276, 2014. ISSN 18737544. doi: 10.1016/j.neuroscience.2013.11.007.

T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of Tricks for Adversarial Training. *ICLR*, 2021.

N. Papernot, P. D. McDaniel, and I. J. Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv preprint arXiv:1605.07277*, 2016.

N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS '17*, 2017. ISBN 9781450349444. doi: 10.1145/3052973.3053009.

A. Pasupathy and C. E. Connor. Responses to contour features in macaque area V4. *Journal of Neurophysiology*, 82(5), 1999. ISSN 00223077. doi: 10.1152/jn.1999.82.5.2490.

H. Paugam-Moisy, R. Martinez, and S. Bengio. Delay learning and polychronization for reservoir computing. *Neurocomputing*, 2008. ISSN 09252312. doi: 10.1016/j.neucom.2007.12.027.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Ã. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011. ISSN 15324435.

G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, 2019.

Z. Pezzementi, C. Reyda, and G. D. Hager. Object mapping, recognition, and localization from tactile geometry. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2011. doi: 10.1109/ICRA.2011.5980363.

R. Pogodin, Y. Mehta, T. Lillicrap, and P. Latham. Towards Biologically Plausible Convolutional Networks. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

W. Pohl. Dissociation of spatial discrimination deficits following frontal and parietal lesions in monkeys. *Journal of Comparative and Physiological Psychology*, 82(2), 1973. ISSN 00219940. doi: 10.1037/h0033922.

F. Pouille and M. Scanziani. Enforcement of temporal fidelity in pyramidal cells by somatic feed-forward inhibition. *Science*, 293(5532), 2001. ISSN 00368075. doi: 10.1126/science.1060342.

P. Quaglio, A. Yegenoglu, E. Torre, D. M. Endres, and S. Grün. Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with SPADE.

Frontiers in Computational Neuroscience, 11, 2017. ISSN 16625188. doi: 10.3389/fncom.2017.00041.

A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning*, 2021.

A. Radhakrishnan, M. Belkin, and C. Uhler. Overparameterized neural networks implement associative memory. *Proceedings of the National Academy of Sciences of the United States of America*, 117(44), 2020. ISSN 10916490. doi: 10.1073/pnas.2005013117.

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv*, 4 2022.

M. Ranzato. On Learning Where To Look. *arXiv preprint arXiv:1405.5488*, 2014.

C. Rao, T. Ikeda, T. Nakata, and H. Liu. Owl-inspired leading-edge serrations play a crucial role in aerodynamic force production and sound suppression. *Bioinspiration and Biomimetics*, 12(4), 2017. ISSN 17483190. doi: 10.1088/1748-3190/aa7013.

R. P. N. Rao and D. H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2 (1):79–87, 1999. ISSN 10976256. doi: 10.1038/4580.

J. Rauber, W. Brendel, and M. Bethge. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017. URL http://arxiv.org/abs/1707.04131.

K. Rayner and A. Pollatsek. Is visual information integrated across saccades? *Perception & Psychophysics*, 34(1), 1983. ISSN 00315117. doi: 10.3758/BF03205894.

L. Reddy and R. VanRullen. Spacing affects some but not all visual searches: Implications for theories of attention and crowding. *Journal of Vision*, 7(2), 2007. ISSN 15347362. doi: 10.1167/7.2.3.

W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In *Sensory Communication*. 1961.

W. Reichardt and M. Egelhaaf. Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly. *Biological Cybernetics*, 58(5), 1988. ISSN 03401200. doi: 10.1007/BF00363937.

D. P. Reichert and T. Serre. Neuronal synchrony in complex-valued deep networks. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

P. Reinagel and R. C. Reid. Temporal coding of visual information in the thalamus. *The Journal of Neuroscience*, 2000. ISSN 0270-6474. doi: 20/14/5392[pii].

M. Riesenhuber and T. Poggio. Are cortical models really bound by the 'binding problem'?, 1999a. ISSN 08966273.

M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11), 1999b. ISSN 10976256. doi: 10.1038/14819.

E. T. Rolls. Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 335(1273), 1992. ISSN 09628436. doi: 10.1098/rstb.1992.0002.

E. T. Rolls and T. Milward. A model of invariant object recognition in the visual system: Learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural Computation*, 2000. ISSN 08997667. doi: 10.1162/089976600300014845.

E. T. Rolls, A. Treves, M. J. Tovee, and S. Panzeri. Information in the neuronal representation of individual stimuli in the primate temporal visual cortex. *Journal of Computational Neuroscience*, 4(4), 1997. ISSN 09295313. doi: 10.1023/A:1008899916425.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (including subseries Lec-*

ture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. ISBN 9783319245737. doi: 10.1007/978-3-319-24574-4{_}28.

J. Rony, L. G. Hafemann, L. S. Oliveira, I. Ben Ayed, R. Sabourin, and E. Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, 2019. doi: 10.1109/CVPR.2019.00445.

F. Rosenblatt. *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC, 1961.

D. Rosenbluth and J. M. Allman. The effect of gaze angle and fixation distance on the responses of neurons in V1, V2, and V4. *Neuron*, 33(1), 2002. ISSN 08966273. doi: 10.1016/S0896-6273(01)00559-1.

C. Rossant, S. Leijon, A. K. Magnusson, and R. Brette. Sensitivity of noisy neurons to coincident inputs. *Journal of Neuroscience*, 31(47), 2011. ISSN 02706474. doi: 10.1523/JNEUROSCI.2482-11.2011.

D. Rumelhart, J. McClelland, and PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Vol. 1)*. MIT Press, 1986.

E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel. A Simple Way to Make Neural Networks Robust Against Diverse Image Corruptions. In *Lecture Notes in Computer Science*, volume 12348 LNCS, 2020.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 2015. ISSN 15731405. doi: 10.1007/s11263-015-0816-y.

E. Russo and D. Durstewitz. Cell assemblies at multiple time scales with arbitrary lag constellations. *eLife*, 6:1–31, 2017. ISSN 2050084X. doi: 10.7554/eLife.19428.

S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017.

T. A. Salthouse and C. L. Ellis. Determinants of eye-fixation duration. *The American journal of psychology*, 93(2), 1980. ISSN 00029556. doi: 10.2307/1422228.

T. Salvatori, Y. Song, Y. Hong, L. Sha, S. Frieder, Z. Xu, R. Bogacz, and T. Lukasiewicz. Associative Memories via Predictive Coding. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

T. Salvatori, L. Pinchetti, B. Millidge, Y. Song, T. Bao, R. Bogacz, and T. Lukasiewicz. Learning on Arbitrary Graph Topologies via Predictive Coding. *arXiv*, (arXiv:2201.13180), 2022.

P. Samangouei, M. Kabkab, and R. Chellappa. Defense-Gan: Protecting classifiers against adversarial attacks using generative models. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot Learning with Memory-Augmented Neural Networks. *NeurIPS 2016 Deep Learning Symposium*, arXiv:1605.06065, 2016.

A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 2009. ISSN 10459227. doi: 10.1109/TNN.2008.2005605.

M. Schak and A. Gepperth. A Study on Catastrophic Forgetting in Deep LSTM Networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11728 LNCS, 2019. doi: 10.1007/978-3-030-30484-3{_}56.

L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on MNIST. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

M. Seeland and P. Mäder. Multi-view classification with convolutional neural networks. *PLoS ONE*, 16(1 January), 2021. ISSN 19326203. doi: 10.1371/journal.pone.0245230.

B. D. Semple, K. Blomgren, K. Gimlin, D. M. Ferriero, and L. J. Noble-Haeusslein. Brain development in rodents and humans: Identifying benchmarks of maturation and vulnerability to injury across species. *Progress in Neurobiology*, 106-107, 2013. ISSN 03010082. doi: 10.1016/j.pneurobio.2013.04.001.

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE TPAMI*, 2007. doi: 10.1109/TPAMI.2007.56.

K. Seymour, C. W. Clifford, N. K. Logothetis, and A. Bartels. The Coding of Color, Motion, and Their Conjunction in the Human Visual Cortex. *Current Biology*, 19(3), 2009. ISSN 09609822. doi: 10.1016/j.cub.2008.12.050.

T. M. Shackleton, B. C. Skottun, R. H. Arnott, and A. R. Palmer. Interaural time difference discrimination thresholds for single neurons in the inferior colliculus of guinea pigs. *Journal of Neuroscience*, 23(2), 2003. ISSN 02706474. doi: 10.1523/jneurosci.23-02-00716.2003.

M. N. Shadlen and J. A. Movshon. Synchrony unbound: A critical evaluation of the temporal binding hypothesis. *Neuron*, 24(1):67–77, 1999. ISSN 08966273. doi: 10.1016/S0896-6273(00)80822-3.

M. N. Shadlen and W. T. Newsome. Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4(4), 1994. ISSN 09594388. doi: 10.1016/0959-4388(94)90059-0.

A. Shafahi, A. Ghiasi, F. Huang, and T. Goldstein. Label Smoothing and Logit Squeezing: A Replacement for Adversarial Training? *arXiv preprint arXiv:1910.11585*, 10 2019. URL http://arxiv.org/abs/1910.11585.

C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 1948. ISSN 15387305. doi: 10.1002/j.1538-7305.1948.tb01338.x.

P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, volume 2, 2018. doi: 10.18653/v1/n18-2074.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, 2014.

C. Singh, B. Yu, and W. James Murdoch. Hierarchical interpretations for neural network predictions. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

S. Sodhani, S. Chandar, and Y. Bengio. Towards training recurrent neural networks for lifelong learning. *Neural Computation*, 2020. ISSN 23318422.

L. Sokoloff. Measurement of local cerebral glucose utilization and its relation to local functional activity in the brain. In *Advances in Experimental Medicine and Biology*, volume 291, 1991. doi: 10.1007/978-1-4684-5931-9{_}4.

S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9), 2000. ISSN 10976256. doi: 10.1038/78829.

S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. In *PLoS Biology*, volume 3, 2005. doi: 10.1371/journal.pbio.0030068.

Y. Song, S. Nowozin, N. Kushman, T. Kim, and S. Ermon. PixelDefend: Leveraging generative models to understand and defend against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

Y. Song, T. Lukasiewicz, Z. Xu, and R. Bogacz. Can the brain do backpropagation? -

Exact implementation of backpropagation in predictive coding networks. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 2014. ISSN 15337928.

R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *Advances in Neural Information Processing Systems*, 2015.

H. Stensola, T. Stensola, T. Solstad, K. Frøland, M. B. Moser, and E. I. Moser. The entorhinal grid map is discretized. *Nature*, 492(7427), 2012. ISSN 00280836. doi: 10.1038/nature11649.

C. Stephenson, J. Feather, S. Padhy, O. Elibol, H. Tang, J. McDermott, and S. Y. Chung. Untangling in invariant speech recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

I. H. Stevenson and K. P. Kording. How advances in neural recording affect data analysis. In *Nature Neuroscience*, volume 14, 2011. doi: 10.1038/nn.2731.

M. Stimberg, R. Brette, and D. F. Goodman. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8, 2019. ISSN 2050084X. doi: 10.7554/eLife.47314.

C. Stringer, M. Pachitariu, N. Steinmetz, M. Carandini, and K. D. Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765), 2019. ISSN 14764687. doi: 10.1038/s41586-019-1346-5.

S. M. Stringer, G. Perry, E. T. Rolls, and J. H. Proske. Learning invariant object recognition in the visual system with continuous transformations. *Biological Cybernetics*, 2006. ISSN 03401200. doi: 10.1007/s00422-005-0030-z.

D. Stutz, M. Hein, and B. Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.00714.

C. Summerfield, F. Luyckx, and H. Sheahan. Structure learning and the posterior

parietal cortex. *Progress in Neurobiology*, 184, 2020. ISSN 18735118. doi: 10.1016/j.pneurobio.2019.101717.

C. Summers and M. J. Dinneen. Improved Adversarial Robustness via Logit Regularization Methods. *arXiv preprint arXiv:1906.03749*, 6 2019. URL http://arxiv.org/abs/1906.03749.

W. A. Suzuki. Neuroanatomy of the monkey entorhinal, perirhinal and parahippocampal cortices: Organization of cortical inputs and interconnections with amygdala and striatum. *Seminars in Neuroscience*, 8(1), 1996. ISSN 10445765. doi: 10.1006/smns.1996.0002.

H. A. Swadlow. Efferent neurons and suspected interneurons in motor cortex of the awake rabbit: Axonal properties, sensory receptive fields, and subthreshold synaptic inputs. *Journal of Neurophysiology*, 71(2), 1994. ISSN 00223077. doi: 10.1152/jn.1994.71.2.437.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, 2016. doi: 10.1109/CVPR.2016.308.

T. Tadros, R. Ramyaa, G. P. Krishnan, and M. Bazhenov. Biologically inspired sleep algorithm for increased generalization and adversarial robustnes in deep neural networks. *ICLR*, 2020.

N. Takahashi, T. Sasaki, W. Matsumoto, N. Matsuki, and Y. Ikegaya. Circuit topology for synchronizing neurons in spontaneously active networks. *Proceedings of the National Academy of Sciences of the United States of America*, 107(22), 2010. ISSN 00278424. doi: 10.1073/pnas.0914594107.

J. W. Tanaka and M. J. Farah. Parts and Wholes in Face Recognition. *The Quarterly*

Journal of Experimental Psychology Section A, 46(2), 1993. ISSN 14640740. doi: 10.1080/14640749308401045.

K. Tanaka. Inferotemporal cortex and object vision. *Annual Review of Neuroscience*, 19, 1996. ISSN 0147006X. doi: 10.1146/annurev.ne.19.030196.000545.

K. Tanaka, H. A. Saito, Y. Fukada, and M. Moriya. Coding visual images of objects in the inferotemporal cortex of the macaque monkey. *Journal of Neurophysiology*, 66(1), 1991. ISSN 00223077. doi: 10.1152/jn.1991.66.1.170.

T. Tanay and L. Griffin. A Boundary Tilting Persepective on the Phenomenon of Adversarial Examples. *arXiv preprint arXiv:1608.07690*, 8 2016. URL http://arxiv.org/abs/1608.07690.

D. W. Tank and J. J. Hopfield. Neural computation by concentrating information in time. *Proceedings of the National Academy of Sciences of the United States of America*, 84 (7), 1987. ISSN 00278424. doi: 10.1073/pnas.84.7.1896.

F. Theunissen and J. P. Miller. Temporal encoding in nervous systems: A rigorous definition. *Journal of Computational Neuroscience*, 2(2), 1995. ISSN 09295313. doi: 10.1007/BF00961885.

A. M. Thomson, J. Deuchars, and D. C. West. Large, deep layer pyramid-pyramid single axon EPSPs in slices of rat motor cortex display paired pulse and frequency-dependent depression, mediated presynaptically and self-facilitation, mediated postsynaptically. *Journal of Neurophysiology*, 70(6), 1993. ISSN 00223077. doi: 10.1152/jn.1993.70.6.2354.

R. A. Tikidji-Hamburyan and C. C. Canavier. Shunting inhibition improves synchronization in heterogeneous inhibitory interneuronal networks with type 1 excitability whereas hyperpolarizing inhibition is better for type 2 excitability. *eNeuro*, 7(3), 2020. ISSN 23732822. doi: 10.1523/ENEURO.0464-19.2020.

E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4), 1948. ISSN 0033295X. doi: 10.1037/h0061626.

F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. The Space of Transferable Adversarial Examples. *arXiv preprint arXiv:1704.03453*, 2017.

F. Tramèr, J. Behrmann, N. Carlini, N. Papernot, and J. H. Jacobsen. Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations. In *37th International Conference on Machine Learning, ICML 2020*, volume PartF168147-13, 2020a.

F. Tramèr, N. Carlini, and W. Brendel. On adaptive attacks to adversarial example defenses. *arXiv*, 2020b. ISSN 23318422.

A. Treisman. The binding problem. *Current Opinion in Neurobiology*, 6(2), 4 1996. ISSN 09594388. doi: 10.1016/S0959-4388(96)80070-5.

A. Treisman. Feature binding, attention and object perception. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 1998. ISSN 09628436. doi: 10.1098/rstb.1998.0284.

A. Treisman. Solutions to the Binding Problem : Progress through Controversy and Convergence. *Neuron*, 24(1):105–125, 1999. ISSN 08966273. URL http://linkinghub.elsevier.com/retrieve/pii/S0896627300808260.

A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 1980. ISSN 00100285. doi: 10.1016/0010-0285(80)90005-5.

A. Treisman and R. Paterson. Emergent features, attention, and object perception. *Journal of Experimental Psychology: Human Perception and Performance*, 1984. ISSN 00961523. doi: 10.1037/0096-1523.10.1.12.

A. Treisman and J. Souther. Search Asymmetry. A Diagnostic for Preattentive Processing of Separable Features. *Journal of Experimental Psychology: General*, 114(3), 1985. ISSN 00963445. doi: 10.1037/0096-3445.114.3.285.

Y. Trotter and S. Celebrini. Gaze direction controls response gain in primary visual-cortex neurons. *Nature*, 398(6724), 1999. ISSN 00280836. doi: 10.1038/18444.

T. W. Troyer, A. E. Krukowski, N. J. Priebe, and K. D. Miller. Contrast-Invariant Orientation Tuning in Cat Visual Cortex: Thalamocortical Input Tuning and Correlation-Based

Intracortical Connectivity. *The Journal of Neuroscience*, 1998. ISSN 0270-6474. doi: 10.1523/jneurosci.18-15-05908.1998.

Y. Tsal. Do Illusory Conjunctions Support the Feature Integration Theory? A Critical Review of Theory and Findings. *Treisman & Gelade Treisman & Schmidt Nis-sen*, 1989. ISSN 0096-1523. doi: 10.1037/0096-1523.15.2.394.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness May Be at Odds with Accuracy. *ICLR*, 2019.

C. Tsvetkov, G. Malhotra, B. Evans, and J. Bowers. Adding biological constraints to deep neural networks reduces their capacity to learn unstructured data. *Proceedings of the 42nd Annual Conference of the Cognitive Science Society*, 2020.

S. Tuli, I. Dasgupta, E. Grant, and T. Griffiths. Are Convolutional Neural Networks or Transformers more like human vision? *arXiv*, arXiv:2105.07197, 2021.

W. M. Usrey, J. M. Alonso, and R. C. Reid. Synaptic interactions between thalamic inputs to simple cells in cat visual cortex. *Journal of Neuroscience*, 20(14), 2000. ISSN 02706474. doi: 10.1523/jneurosci.20-14-05461.2000.

R. Van Der Meij and B. Voytek. Uncovering Neuronal Networks Defined by Consistent between-Neuron Spike Timing from Neuronal Spike Recordings. *eNeuro*, 10, 2018. doi: 10.1523/ENEURO.0379-17.2018.

J. P. H. van Santen and G. Sperling. Temporal covariance model of human motion perception. *Journal of the Optical Society of America A*, 1(5), 1984. ISSN 1084-7529. doi: 10.1364/josaa.1.000451.

S. van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

R. VanRullen. Binding hardwired versus on-demand feature conjunctions. *Visual Cognition*, 17(1-2), 2009. ISSN 13506285. doi: 10.1080/13506280802196451.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Å. Kaiser, and

I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

C. von der Malsburg. The Correlation Theory of Brain Function. Technical report, MPI Biophysical Chemistry, Internal Report 81-2, 1981.

C. Von Der Malsburg. The what and why of binding: The modeler's perspective. *Neuron*, 24(1):95–104, 1999. ISSN 08966273. doi: 10.1016/S0896-6273(00)80825-9.

M. Vuyyuru, A. Banburski, N. Pant, and T. Poggio. Biologically Inspired Mechanisms for Adversarial Robustness. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychological Bulletin*, 138(6), 2012. ISSN 00332909. doi: 10.1037/a0029333.

G. Wallis and E. T. Rolls. Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51(2), 1997. ISSN 03010082. doi: 10.1016/S0301-0082(96)00054-8.

B. Wang, L. Shang, C. Lioma, X. Jiang, H. Yang, Q. Liu, and J. G. Simonsen. On Position Embeddings in BERT. *ICLR*, 2021.

Q. Wang, P. Cavanagh, and M. Green. Familiarity and pop-out in visual search. *Perception & Psychophysics*, 56(5), 1994. ISSN 00315117. doi: 10.3758/BF03206946.

D. Warde-Farley and I. Goodfellow. 11 adversarial perturbations of deep neural networks. In *Perturbations, Optimization, and Statistics*, page 311. 2016.

C. Whitney. How the brain encodes the order of letters in a printed word: The seriol model and selective literature review. *Psychonomic Bulletin and Review*, 8(2), 2001. ISSN 10699384. doi: 10.3758/BF03196158.

J. Whittington, J. Warren, and T. Behrens. Relating Transformers to Models and Neural Representations of the Hippocampal Formation. *International Conference on Learning Representations (ICLR)*, 2022.

J. C. Whittington and R. Bogacz. Theories of Error Back-Propagation in the Brain. *Trends in Cognitive Sciences*, 23(3), 2019. ISSN 1879307X. doi: 10.1016/j.tics.2018.12.005.

J. C. Whittington, T. H. Muller, C. Barry, S. Mark, and T. E. Behrens. Generalisation of structural knowledge in the hippocampal-entorhinal system. In *Advances in Neural Information Processing Systems*, volume 2018-December, 2018.

J. C. Whittington, T. H. Muller, S. Mark, G. Chen, C. Barry, N. Burgess, and T. E. Behrens. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183, 11 2020. ISSN 00928674. doi: 10.1016/j.cell.2020.10.024.

A. H. Williams, B. Poole, N. Maheswaranathan, A. K. Dhawale, T. Fisher, C. D. Wilson, D. H. Brann, E. M. Trautmann, S. Ryu, R. Shusterman, D. Rinberg, B. P. Ölveczky, K. V. Shenoy, and S. Ganguli. Discovering Precise Temporal Patterns in Large-Scale Neural Recordings through Robust and Interpretable Time Warping. *Neuron*, 105(2), 2020. ISSN 10974199. doi: 10.1016/j.neuron.2019.10.020.

C. Wloka, I. Kotseruba, and J. K. Tsotsos. Active Fixation Control to Predict Saccade Sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00336.

J. M. Wolfe. Forty years after feature integration theory: An introduction to the special issue in honor of the contributions of Anne Treisman. *Attention, Perception, and Psychophysics*, 82(1), 2020. ISSN 1943393X. doi: 10.3758/s13414-019-01966-3.

J. M. Wolfe and J. S. DiMase. Do intersections serve as basic features in visual search? *Perception*, 32(6), 2003. ISSN 03010066. doi: 10.1068/p3414.

J. M. Wolfe, S. R. Friedman-Hill, and A. B. Bilsky. Parallel processing of part-whole information in visual search tasks. *Perception & Psychophysics*, 55(5), 1994. ISSN 00315117. doi: 10.3758/BF03205311.

A. Wong and A. Yuille. One shot learning via compositions of meaningful patches. *Proceedings of the IEEE International Conference on Computer Vision (2015)*, 2015.

C. Xiao, P. Zhong, and C. Zheng. Enhancing Adversarial Defense by k-Winners-Take-All. *International Conference on Learning Representations 2020*, 2020.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 8 2017. URL http://arxiv.org/abs/1708.07747.

C. Xu, J. Yang, H. Lai, J. Gao, L. Shen, and S. Yan. UP-CNN: Un-pooling augmented convolutional neural network. *Pattern Recognition Letters*, 2019. ISSN 01678655. doi: 10.1016/j.patrec.2017.08.007.

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 111(23), 2014. ISSN 10916490. doi: 10.1073/pnas.1403112111.

A. L. Yarbus. Eye Movements During Perception of Complex Objects. In *Eye Movements and Vision*. 1967.

X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE transactions on neural networks and learning systems*, 2019. ISSN 21622388. doi: 10.1109/TNNLS.2018.2886017.

A. M. Zador. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, 10(1), 2019. ISSN 20411723. doi: 10.1038/s41467-019-11786-6.

S. Zagoruyko and N. Komodakis. Wide Residual Networks. In *British Machine Vision Conference 2016, BMVC 2016*, volume 2016-September, 2016. doi: 10.5244/C.30.87.

P. Zamarashkina, D. V. Popovkina, and A. Pasupathy. Timing of response onset and offset in macaque V4: stimulus and task dependence. *Journal of neurophysiology*, 123 (6), 2020. ISSN 15221598. doi: 10.1152/jn.00586.2019.

T. P. Zanos, P. J. Mineault, K. T. Nasiotis, D. Guitton, and C. C. Pack. A Sensorimotor Role for Traveling Waves in Primate Visual Cortex. *Neuron*, 85(3), 2015. ISSN 10974199. doi: 10.1016/j.neuron.2014.12.043.

V. Zantedeschi, M. I. Nicolae, and A. Rawat. Efficient defenses against adversarial atacks. In *AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017*, 2017. doi: 10.1145/3128572.3140449.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014. ISBN 9783319105895. doi: 10.1007/978-3-319-10590-1-53.

S. M. Zeki. Functional specialisation in the visual cortex of the rhesus monkey. *Nature*, 274(5670), 1978. ISSN 00280836. doi: 10.1038/274423a0.

F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *34th International Conference on Machine Learning, ICML 2017*, volume 8, 2017.

H. Zhang, A. J. Watrous, A. Patel, and J. Jacobs. Theta and Alpha Oscillations Are Traveling Waves in the Human Neocortex. *Neuron*, 98(6), 2018. ISSN 10974199. doi: 10.1016/j.neuron.2018.05.019.

R. Zhang. Making convolutional networks shift-invariant again. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, 2019.

H. Zhou, H. S. Friedman, and R. von der Heydt. Coding of border ownership in monkey visual cortex. *The Journal of Neuroscience*, 20(17):6594–6611, 2000. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2797-12.2013.

Z. Zhou and C. Firestone. Humans can decipher adversarial images. *Nature Communications*, 10(1), 2019. ISSN 20411723. doi: 10.1038/s41467-019-08931-6.

D. Zipser and R. A. Andersen. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331(6158), 1988. ISSN 00280836. doi: 10.1038/331679a0.