Causal and Identifiable Deep Representation Learning



Marian Longa

Linacre College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023

Shallow men believe in luck.
Strong men believe in cause and effect.

– Ralph Waldo Emerson

Acknowledgements

First of all, I would like to thank my advisor, João Henriques, without whom none of this would be possible. From the time I worked with him as an undergraduate until now, João has always been very supportive, gave me space to explore my own ideas, and has taught me the value of pursuing difficult problems.

Next, I would like to thank my friends and colleagues, Subha Choudhury, Edgar Sucar, Tom Jakab, Eldar Insafutdinov, Lorenza Prospero, Andreea Oncescu, and many others from VGG and Linacre College, who have supported me during my DPhil and made it so much more fun.

Special thanks to Ashish Thandavan and Jenny Hu for taking care of our servers and our lab.

Finally, I would like to thank my parents and family for their support.

Abstract

Deep Learning has been responsible for many of the recent artificial intelligence (AI) successes, from text generation, image understanding, protein structure prediction, to superhuman performance in playing chess. Despite their successes, most current deep learning methods have difficulties with generalisation beyond the training distribution, are often not interpretable or explainable, possess few learning guarantees, and are often not robust to environmental changes. As AI is being adopted in more and more parts of society, it is important to understand and address these limitations.

One way to address these issues is by applying the framework of causality to deep learning, in what has become the emerging field of causal representation learning. In causal learning, the emphasis is on using data to learn causal variables and relationships instead of pure statistical associations, as this allows for improved interpretability, transfer, and reasoning via the framework of do-calculus.

In this thesis we take inspiration from causality and identifiability to develop deep learning methods for visual data that are interpretable, possess learning guarantees, and generalise beyond the training distribution. We do this by designing deep learning systems that employ various assumptions, such as assumptions about the environment (e.g. video with a static background), assumptions about the latent space (e.g. having one or two variables, or using a causal graph), restrictions on the causal mechanism function classes (e.g. linear functions), and properties of the network architectures (e.g. equivariance to certain transformations). Using these assumptions allows us to prove that the models are guaranteed to learn the true latent variables (up to some transformations), show that the models successfully generalise beyond the training distribution by intervening on the latent representation and generating realistic videos never observed at training time, and demonstrate that the learned representation is easily interpretable and explainable.

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.1.1 Successes of Deep Learning . . . 1

1.1.2 Limitations of Deep Learning . . . 2

1.1.3 Causality . . . 4

1.1.4 Causal Representation Learning . . . 5

1.1.5 Causal vs. Classical Learning . . . 6

1.1.6 Aim of This Thesis . . . 8

1.2 Outline . . . 8

1.3 Contributions . . . 9

1.3.1 Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors . . . 9

1.3.2 Interpretable Representation Learning from Videos using Nonlinear Priors . . . 10

1.3.3 Unsupervised Object Detection with Theoretical Guarantees 11

1.4 List of Publications . . . 12

2 Literature Review **13**

2.1 Causality . . . 13

2.2 Representation Learning . . . 15

2.3 Causal Representation Learning . . . 16

2.4 Equivariances and Invariances . . . 17

2.5 Intuitive Physics . . . 18

3 Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors **19**

3.1 Introduction . . . 19

3.2 Method . . . 21

3.2.1 Variational autoencoder . . . 22

3.2.2 Additive noise model prior . . . 23

3.2.3 Probabilistic edge penalisation . . . 24

3.2.4 Prediction loss . . . 24

3.3 Experiments . . . 25

3.4 Conclusion . . . 30

4 Interpretable Representation Learning from Videos using Nonlinear Priors **33**
4.1 Introduction . . . 33
4.2 Method . . . 35
4.2.1 Optimisation Objective . . . 36
4.2.2 Posterior GMM . . . 36
4.2.3 Prior GMM . . . 36
4.2.4 KL Divergence . . . 38
4.2.5 Multivariate case . . . 39
4.3 Experiments . . . 40
4.3.1 Implementation . . . 41
4.3.2 Mass on a Spring . . . 42
4.3.3 Pendulum . . . 43
4.3.4 Falling Object . . . 45
4.3.5 Pulsar . . . 46
4.4 Discussion . . . 47
4.5 Conclusion . . . 48

5 Unsupervised Object Detection with Theoretical Guarantees **51**
5.1 Introduction . . . 51
5.2 Method . . . 53
5.2.1 Autoencoder with CNN Encoder and Decoder . . . 54
5.2.2 From Encoder Feature Maps to Latent Variables . . . 55
5.2.3 From Latent Variables to Decoder Feature Maps . . . 56
5.3 Theoretical Results . . . 57
5.4 Experimental Results . . . 62
5.4.1 Synthetic Experiments . . . 64
5.4.2 CLEVR Experiments . . . 65
5.5 Discussion . . . 67
5.6 Conclusion . . . 69

6 Conclusion **71**
6.1 Summary . . . 71
6.2 Limitations and Challenges . . . 72
6.3 Future Work . . . 73

References **75**

Appendices

A Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors **87**

A.1 Proof of DAG maximality 87

B Identifiable Representation Learning from Architecture Equivariances **89**

B.1 Proof of Theorem 5.3.1 89

B.2 Theoretical Results for Multiple Object Sizes 91

B.3 Experiment Training Details 92

B.3.1 Synthetic Experiments 92

B.3.2 CLEVR Experiments 94

B.4 CLEVR Experiments with Different Shapes 96

B.5 Experiments with Real Videos 97

B.5.1 Traffic Data 97

B.5.2 Mini Pool 97

C Sources of Figures **99**

1 Introduction

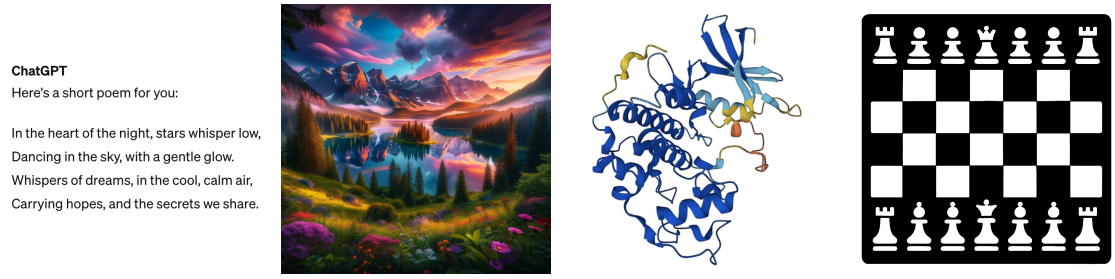
Contents

1.1 Motivation **1**
1.1.1 Successes of Deep Learning 1
1.1.2 Limitations of Deep Learning 2
1.1.3 Causality 4
1.1.4 Causal Representation Learning 5
1.1.5 Causal vs. Classical Learning 6
1.1.6 Aim of This Thesis 8
1.2 Outline **8**
1.3 Contributions **9**
1.3.1 Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors 9
1.3.2 Interpretable Representation Learning from Videos using Nonlinear Priors 10
1.3.3 Unsupervised Object Detection with Theoretical Guarantees 11
1.4 List of Publications **12**

1.1 Motivation

1.1.1 Successes of Deep Learning

Deep learning has been responsible for many of the recent AI successes, ranging from realistic text generation and understanding using GPT [1], image generation from text using DALL-E [2], protein structure prediction with AlphaFold [3], to



(a) Text generation **(b)** Image generation **(c)** Protein prediction **(d)** Playing chess

Figure 1.1: Recent advances in deep learning, including (a) text generation and understanding using GPT, (b) text to image generation using DALL-E, (c) protein structure prediction using AlphaFold, and (d) superhuman performance at playing chess using MuZero.

playing games like chess and Go with MuZero [4] (fig. 1.1). Typically, deep learning involves fitting neural network models to large datasets consisting of many input and output pairs (for example, to automatically classify objects in images, a dataset might consist of pairs of images and the corresponding object descriptions). Before deep learning, the field of AI was dominated by classical machine learning methods, however these have been surpassed by more powerful deep learning methods in the last decade [5]. The rise of deep learning has been enabled by the use of very large datasets [6], increasing computing power [7], as well as better models [1, 2, 8, 9], and is poised to continue into the future. To ensure safe and reliable adoption of AI throughout society, it is thus important to understand the limitations of deep learning.

1.1.2 Limitations of Deep Learning

As deep learning is being adopted in more and more areas of society, it is important to understand its limitations and potential dangers (fig. 1.2). Firstly, most current deep learning methods are not interpretable, making them difficult to adopt in areas such as healthcare or law, where it is essential to know how machines reach their conclusions (fig. 1.2a). For example, to deploy a deep learning based system for medical diagnosis, we would like to design methods whose working is inherently understandable to the medical personnel beyond the current *post hoc* methods of analysing the models' learned weights.



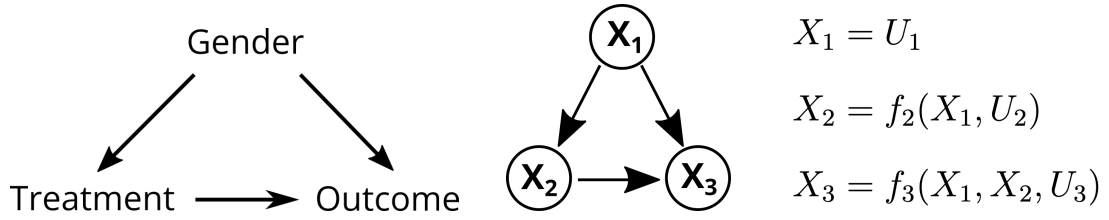
(a) Explainability **(b)** Generalisation **(c)** Learning guarantees **(d)** Robustness

Figure 1.2: Challenges of current deep learning systems, including (a) explainability of machine learning models in medical applications, (b) out-of distribution generalisation, such as detecting objects in a novel context, (c) obtaining guarantees about deep learning algorithms for autonomous driving, and (d) robustness of machine learning algorithms to environmental factors such as lighting.

The second issue is the problem of out-of-distribution generalisation, that is, generalisation from the training distribution to a test distribution that is different. While one may argue that current methods are already good at generalising to new data, in practice they often fail when presented with a dataset that is truly different to their training set. For example, object detectors trained on the COCO dataset [10] might not recognise a cow lying on sand because they might not have seen this combination during training (fig. 1.2b). We would thus like to design methods that generalise to truly novel situations.

The third problem is that current deep learning methods have few theoretical guarantees, making them risky to apply to sensitive domains such as autonomous driving, where it is critical to succesfully detect objects such as pedestrians or other cars (fig. 1.2c). While one may argue that current deep learning based object detection methods such as SAM [11] or CUTLER [12] already work very well, they are not *guaranteed* to do so in 100% of the cases. We would thus like to design deep learning methods with strong theoretical guarantees.

Finally, current methods struggle with robustness and transfer to different domains, for example being able to detect objects under different environmental conditions such as lighting or different viewpoints (fig. 1.2d). Again, one might argue that modern methods already possess a certain degree of robustness, such as robustness to weather and lighting conditions in autonomous driving models.



(a) An example of a causal system **(b)** Causal graph **(c)** Structural causal model

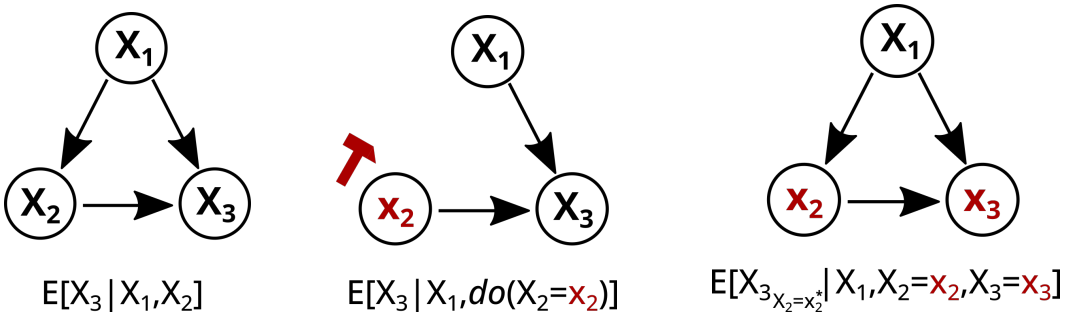
Figure 1.3: Structures used to represent the causal system in (a) where the patient's gender affects both the treatment they receive and the outcome of the treatment, including (b) a causal graph consisting of nodes representing variables and directed edges representing causal relationships, and (c) a structural causal model describing the dependencies between variables X via functional mechanisms f and independent noises U.

However, the models may not be robust to rare cases such as driving with snow in Florida if these have not been observed at training time. We would therefore like to develop methods that are robust, by their design, to different kinds of perturbations, instead of their robustness purely coming from the variety in their training data.

To make AI safer and more trustworthy, it is thus important to address all of these challenges.

1.1.3 Causality

To help address the limitations described in sec. 6.2, there has been recent interest in applying ideas from causality [13] to deep learning. *Causality* refers to the property of the change in one variable (the *cause*) inducing a change in another variable (the *effect*). More formally, causality is often represented using the framework of Structural Causal Models (SCMs) developed by Judea Pearl [13], where a causal system such as the one in fig. 1.3a can be represented by a *causal graph* in fig. 1.3b and a *structural causal model* in fig. 1.3c. In this definition, the graph nodes represent variables (X_i in fig. 1.3), and the graph edges represent causal mechanisms relating the variables (f_i in fig. 1.3). We then say that variable X_i *causes* X_j if the intervention on the variable X_i causes the probability distribution of X_j to change, i.e. $P(X_j|do(X_i = x_1)) \neq P(X_j|do(X_i = x_2))$. In this expression, $do(X = x)$ represents an *intervention*, i.e. fixing the value of X to x (fig. 1.4b), which allows one to reason about the causal effect of one variable on another. One can also



(a) Original **(b)** Intervention **(c)** Counterfactual

Figure 1.4: (a) Original causal graph where one would like to compute the value of X_3, (b) intervention in the graph where one wants to compute the value of X_3 given that the value of X_2 is set to x_2, (c) counterfactual where one would like to compute the value of X_3 for the hypothetical case of $X_2 = x_2^*$, given that one has observed $X_2 = x_2$ and $X_3 = x_3$.

compute *counterfactuals*, which refers to reasoning about hypothetical scenarios of what would the value of one variable had been if another variable had a different value than its observed value (fig. 1.4c).

In classical causality, researchers are typically provided with the variables describing data in a particular domain and aim to answer causal questions about it. These can include determining the causal effect of a particular variable in the system, known as *causal inference*, and determining how the variables are causally related to each other, known as *causal discovery*. For example, in economics, one may want to find the effect of a training program on the employability of the participants [14]. Or, in healthcare, one might be interested in finding out the causal structure that describes how different proteins interact with one another [15].

In the next section we explore how we can apply causality to deep learning.

1.1.4 Causal Representation Learning

To address the issues presented in sec. 6.2, recently researchers have started combining ideas from causality and machine learning, in what has become known as causal representation learning [16] [1]. In this framework, researchers are interested in learning causal representations, i.e. variables and their causal relationships, from

[1]When I started my DPhil in 2019 with the aim to combine deep learning with ideas from causality, causal representation learning was not yet an established field.

high-dimensional data such as images or video. Unlike standard representation learning, causal representation learning aims to learn causal representations instead of pure statistical associations from data. And unlike classical causality, causal representation learning aims to learn the latent variables instead of assuming they are provided *a priori*.

The advantage of learning causal representations is that they provide a way to address the limitations of deep learning described in sec. 6.2. For example, causal graph representations are more intuitively interpretable (fig. 1.2a), they are more robust to environment variations since a change in one variable only affects the causal graph locally (fig. 1.2d), and they are better at out-of-distribution generalisation since they capture causal relationships instead of spurious correlations in the data (fig. 1.2b) [16]. They also allow reasoning about hypothetical scenarios via the use of interventions and counterfactuals on the learned causal graph, something that is not possible in standard representation learning [13].

Because learning both latent variables and their causal relationships from data can be a very underconstrained problem, a central issue in causal representation learning is that of *identifiability*, that is the degree to which the learned variables correspond to their ground truth values. To achieve identifiability, researchers employ various assumptions, such as assumptions on the latent variable distributions, the form of the causal mechanisms, and the types of interventions allowed in the system. Designing methods with provable identifiability guarantees will help address the final issue presented in the previous section (fig. 1.2c).

In the next section we will contrast causal and classical learning methods.

1.1.5 Causal vs. Classical Learning

While classical correlation-based learning performs very well in standard IID[2] settings, causal learning has been shown to outperform it in situations where one has access to prior knowledge such as known invariants, domain labels, or interventions in the system. For example, methods such as Invariant Risk Minimisation (IRM)

[2]Independent and identically distributed.

[17] and Anchor Regression [18] have demonstrated superior generalisation in classification and regression tasks under distribution shifts, Bengio *et al.* [19] achieve better generalisation by meta-learning causal mechanisms, Magliacane *et al.* [20] achieve it via data from different environments, and Kaushik *et al.* [21] use counterfactual data augmentation to improve the robustness of NLP models. However, despite their advantages, causal learning methods often deal with toy data and require strong assumptions.

To demonstrate the benefits of using a causal learning method as opposed to a standard representation learning method, consider the example of a helium balloon rising up in the sky. As the balloon rises, it expands in size due to the lower air pressure at higher altitudes. If one were to train a classical representation learning method on this data, it would correlate the balloon's vertical position with its size, because it has no way of disentangling these two variables. In contrast, in a causal learning method, one might intervene on the system by releasing the balloon from different heights while keeping its starting size constant, and train a model where one variable (height) causes another variable (size). One can then use the learned causal system to interpret the physical process (higher heights result in larger balloons), generalise to novel combinations of heights and balloon sizes (releasing the balloon from a height never observed during training, the model would still predict that it will expand), and make the balloon's height be robust to changes in its size (since there is no causal link from size to height, only vice versa).

One method to distinguish a causal and a correlational representation is to compute a set of conditionals. For example, given the balloon's height H and size S, one can verify the causal relationship $H \to S$ by checking if $P(S|do(H = h)) \neq P(S|do(H = h'))$, and that $S \nrightarrow H$ by checking if $P(H|do(S = s)) = P(H|do(S = s'))$. In contrast, in a classical correlation-based method, both H and S would be correlated, i.e. $H \to S$ (as before), but now also $S \to H$, which can be verified by checking if $P(H|do(S = s)) \neq P(H|do(S = s'))$.

In light of the background presented thus far, in the next section we present the aim of this thesis.

1.1.6 Aim of This Thesis

On a high level, the aim of this thesis is to develop methods and theoretical analysis grounded in causality and identifiability to help improve the current deep learning systems. Causality offers a compelling framework to help move beyond learning statistical associations to learning causal relationships from data, a feat that would make current methods more interpretable, generalisable, and robust [16], thus making them better suited to sensitive applications such as those in healthcare and law.

To do this, we would like to design deep learning systems that learn causal representations from data and that possess identifiability guarantees about their learned representations. In practice, this means developing methods that incorporate various assumptions about the data, about the latent variables, and about the causal mechanisms, in order to obtain a balance between real world applicability on one hand and identifiability guarantees on the other hand. The overall goal would be to obtain methods that learn from real world data such as YouTube videos, possess strong guarantees about the learned representations, learn representations that are transferrable to novel settings and that are robust to environmental changes, as well as representations that allow for reasoning about hypothetical scenarios. In the following chapters we will aim to make progress towards these aims.

1.2 Outline

This thesis is organised into two introductory chapters (chapters 1, 2), 3 chapters with core contributions (chapters 3, 4, 5), and one concluding chapter (chapter 6).

1. Chapter 1 introduces the topic of causal and identifiable representation learning and motivates its importance in the context of current machine learning systems.

2. Chapter 2 reviews literature in relevant areas including causality, representation learning, causal representation learning, intuitive physics, and equivariances.

3. Chapter 3 presents a general unsupervised learning method for learning causal variables and their relationships from image data using an additive noise model prior with edge sparsity, and demonstrates it in a game-based setting.

4. Chapter 4 presents an unsupervised learning method for learning interpretable variables representing a physical property of an object in real physics videos using non-linear priors, which can be intervened to generate realistic counterfactual videos never seen during training.

5. Chapter 5 presents the first unsupervised object detection method that is theoretically guaranteed to recover the true object positions up to quantifiable small shifts, by exploiting equivariances.

6. Chapter 6 summarises the thesis and outlines directions for future work.

1.3 Contributions

The overarching contribution of this thesis is the development of unsupervised learning methods that learn (causal) latent representations of high-dimensional data, such as images or video, that are interpretable, identifiable, and transferable. That is, unlike for most current unsupervised deep learning methods, one can guarantee certain properties of the learned latent variables and have the ability to successfully transfer them to novel settings outside of the training distribution.

1.3.1 Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors

In chapter 3 we present an unsupervised learning method for learning a causal model of image and video data using an additive noise model prior with sparsity penalisation. That is, given a video, the method jointly learns latent variables and their causal relationships representing the video, using a prior on how the variables should be related, and by encouraging sparse causal relationships. This is a very general method that starts with a fully connected graph and over time learns a subset of the variables and mechanisms, removing unnecessary graph edges.

Depending on the strength of the penalisation, this allows for controlling the balance between the simplicity of the graph and the quality of the reconstruction. We apply the method to video game based data with a character moving at a constant speed, and show that given a linear prior, the method learns variables representing the object's position and by intervening on the variables we can succesfully extrapolate the object's position into the future. While general, this model may become very underconstrained, so in the following two chapters we seek to obtain greater identifiability guarantees about the learned representations by focusing on single and multi-variable cases and incorporating additional assumptions.

Concretely, the contributions of this work are:

1. A method for deep learning of causal variables and mechanisms by maximisation of the evidence lower bound (ELBO) of a variational autoencoder (VAE) with an additive noise model (ANM) prior coupled with edge penalisation.

2. Experiments on video game based data demonstrating learning of the correct causal variables and mechanisms.

3. Experiments using the learned causal graph to visualise interventions and extrapolate system behaviour into the future.

1.3.2 Interpretable Representation Learning from Videos using Nonlinear Priors

In chapter 4 we propose a deep learning framework for learning physically interpretable variables from videos by specifying non-linear priors, and use it to generate physics videos of hypothetical scenarios never observed at training time. We do this by using a Variational Auto-Encoder (VAE) and replacing the standard isotropic Gaussian prior with a nonlinear temporal additive noise model (ANM), which can describe a large number of processes (e.g. Newtonian physics). We propose a novel linearization method that constructs a Gaussian Mixture Model (GMM) approximating the prior, and derive a numerically stable Monte Carlo estimate of the KL divergence between the posterior and prior GMMs. We validate the

method on different real-world physics videos including a pendulum, a mass on a spring, a falling object and a pulsar (rotating neutron star). We specify a physical prior for each experiment and show that the correct variables are learned. Once a model is trained, we intervene on it to change different physical variables (such as oscillation amplitude or adding air drag) to generate physically correct videos of hypothetical scenarios that were not observed previously. Finally, we prove that the learned representation is identifiable, which is possible by assuming a single variable and by providing a non-linear causal mechanism prior. In the following chapter we will relax the single-variable assumption.

Concretely, the contributions of this work are:

1. A novel interpretable representation learning framework based on a Variational Auto-Encoder with a nonlinear Additive Noise Model prior, for learning temporal variables from video, and the ability to intervene on the model to produce realistic counterfactual videos.

2. A novel method for approximation of the prior density via local linearizations of the prior, automatically decomposing it into a mixture of non-isotropic Gaussians, as well as a numerically stable and highly parallelizable KL divergence estimate between prior and posterior distributions.

3. Experiments demonstrating the method on 4 real physics experiments, showing that the correct latent variables are learned, and showing the generation of realistic physics videos outside of the training distribution.

1.3.3 Unsupervised Object Detection with Theoretical Guarantees

Unsupervised object detection using deep neural networks is typically a difficult problem with few to no guarantees about the learned representation. In chapter 5 we present the first unsupervised object detection method that is theoretically guaranteed to recover the true object positions up to quantifiable small shifts. We develop an unsupervised object detection architecture and prove that the learned

variables correspond to the true object positions up to small shifts related to the encoder and decoder receptive field sizes, the object sizes, and the widths of the Gaussians used in the rendering process. We perform detailed analysis of how the error depends on each of these variables and perform synthetic experiments validating our theoretical predictions up to a precision of individual pixels. We also perform experiments on CLEVR-based data and show that, unlike current SOTA object detection methods (SAM, CutLER), our method's prediction errors always lie within our theoretical bounds. We hope that this work helps open up an avenue of research into object detection methods with theoretical guarantees.

Concretely, the contributions of this work are:

1. The first unsupervised object detection method that is guaranteed to learn the true object positions up to small shifts.

2. A proof and detailed theoretical analysis of how the maximum position error of the method depends on the encoder and decoder receptive field sizes, object sizes, and widths of the Gaussians used in the rendering process.

3. Synthetic and CLEVR-based experiments validating our theoretical results up to precisions of individual pixels.

1.4 List of Publications

- Chapter 4: Marian Longa and João F. Henriques. "Interpretable Representation Learning from Videos using Nonlinear Priors". In: *35th British Machine Vision Conference 2024 (BMVC 2024)*. (Oral presentation). https://bmva-archive.org.uk/bmvc/2024/papers/Paper_304/paper.pdf

- Chapter 5: Marian Longa and João F. Henriques. "Unsupervised Object Detection with Theoretical Guarantees". In: *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. https://proceedings.neurips.cc/paper_files/paper/2024/file/b9b1d29de259b82c944b03b7322eae45-Paper-Conference.pdf

2 Literature Review

Contents

2.1 Causality **13**
2.2 Representation Learning **15**
2.3 Causal Representation Learning **16**
2.4 Equivariances and Invariances **17**
2.5 Intuitive Physics **18**

This thesis lies on the intersection of the fields of deep learning and causal inference, in what is now known as causal representation learning [16]. As the ideas presented in this thesis cover multiple fields, this chapter will review the relevant areas, starting from classical causality (sec. 2.1) and classical representation learning (sec. 2.2), followed by the recent advances in causal representation learning (sec. 2.3), and concluding with the adjacent areas of equivariances and invariances (sec. 2.4) and intuitive physics (sec. 2.5).

2.1 Causality

In the field of classical causality, researchers are typically provided with variables representing data in a particular domain with the aim of answering causal questions about it. Broadly, causal methods can be categorised into *causal inference* and

causal discovery.

Causal inference deals with estimating the causal effect of a particular variable in a system from data (e.g. the effect of a training program on income and employment status [14]). The two main frameworks for dealing with causal inference are Pearl's structural causal model (SCM) framework [13, 22] and Rubin's potential outcomes framework [23] [24]. The SCM framework consists of a causal graph representing the variables and causal relationships between them, together with a structural equation model describing the functional relationships of how variables depend on each other. The potential outcomes framework typically deals with estimating the causal effect of a variable by comparing the outcome under treatment and outcome without treatment, where only one of the two potential outcomes is observed.

Causal discovery deals with discovering the causal structure of a system from data (e.g. estimating how proteins in a biological system interact with one another [15]). Methods in causal discovery can be broadly categorised into constraint-based methods, score-based methods, and methods based on functional causal models [25]. Constraint-based methods work by testing for independences between variables, leading to different sets of causal structures (e.g. the PC and FCI algorithms [26]). Score-based methods search through graphs by optimising a score function to find the right causal structure (e.g. the GES algorithm [27]). Methods based on functional causal models take advantage of the constraints imposed by a specified functional relationship between the variables to find the right causal structure (e.g. the LiNGAM algorithm [28], and nonlinear algorithms [29, 30]).

This work is inspired by both causal inference and causal discovery, as we would like to both learn the causal relationships between variables and then intervene on the variables to imagine their effects in hypothetical scenarios. However, the main difference from classical causality methods is that our aim is to learn the variables automatically from data, as opposed to assuming they are provided to us a priori, as is the case with the presented methods.

2.2 Representation Learning

The area of representation learning deals with learning representations of data that capture its explanatory factors and that can be used for further supervised tasks such as classification or prediction [31].

There has been a wide range of models used for representation learning in different domains. For example, in computer vision, convolutional neural networks (CNNs) and vision transformers (ViT) [9] have been used for tasks such as object detection, classification, and image segmentation; and normalising flows [32] and diffusion models (such as DALL-E) [2] have been used for image generation. In natural language processing, recurrent neural networks (RNNs) [33], long short term memory networks (LSTMs) [34], and more recently transformer-based models such as BERT [35] and GPT [1] have been used for tasks such as text understanding, translation, and generation. However, many of these models are trained with supervision (e.g. image and label pairs for CNNs), while our aim would be to learn without supervision. One example of an unsupervised learning method is the variational autoencoder (VAE) [36], which has been used for image generation and denoising. In this work we focus on autoencoders as they are an unsupervised learning method with a low-dimensional latent space that is easy to interpret.

An autoencoder is a model used for learning a compressed representation of data by compressing it using an encoder neural network and decompressing it to obtain the original data using a decoder network. A particular type of autoencoder that encodes data into a probability distribution is the variational autoencoder (VAE) [36]. Multiple works have been proposed that are based on the VAE with differently structured priors, where the aim is usually to design a prior that achieves good image reconstructions (data negative log likelihood) on a standard set of image benchmarks. For example, the β-VAE [37] and FactorVAE [38] aim to learn independent latent variables, while Ladder-VAE [39] conditions each variable on the preceding one and NVAE [40] and VDVAE [41] condition each variable on all the previous variables.

2.3 Causal Representation Learning

Causal representation learning (CRL) is a recent field at the intersection of causality and machine learning that aims to learn high-level causal variables from low-level data [16]. Typically, this involves learning both the latent variables as well as their causal relationships (known as *mechanisms*) from high-dimensional data such as images, by making various assumptions in order to obtain a representation that is *identifiable*, i.e. that the learned representation is unique or corresponds to the true underlying solution up to some transformations (such as variable permutations or scaling).

Beginnings of CRL can be traced back to independent component analysis (ICA) [42], a technique in signal processing that recovers independent sources of a given signal up to permutations and scaling, if the sources are independent, if at most one source is Gaussian, and if sources are mixed using a linear transformation [42]. Relaxing the strong linearity assumption, Locatello et al. [43] proves that for a system with independent variables mixed using a nonlinear transformation, it is not possible to recover the true variables. Therefore, to obtain identifiability in nonlinear systems, one needs to make additional assumptions. Xi et al. [44] broadly categorise these assumptions into the following categories: distributional assumptions on an auxiliary variable such as time (e.g. nonlinear ICA / TCL [45]) or an environment variable (e.g. iVAE [46]) or an auxiliary variable (e.g. SlowVAE [47]); assumptions on the mechanisms such as linearity (e.g. linear ICA [42]) or time evolution (e.g. [48]); using multiple views (e.g. [49, 50]); using interventions (e.g. [51–53]); assuming sparsity of mechanisms (e.g. [54–56]).

In this work we achieve identifiability in multiple different ways. In the video experiments in chapter 3 we assume linear causal mechanisms over multiple time steps, in chapter 4 we assume a single latent variable conditioned on time, and in chapter 5 we design a new equivariance-based architecture that ensures that we learn identifiable object positions. Also, unlike most other works, our aim is to move from synthetic data towards real videos (especially in chapters 4 and 5).

2.4 Equivariances and Invariances

Equivariances and invariances are important concepts in deep learning. Equivariance refers to the property of a function (such as a deep network) that applying some transformation to the input will result in the same transformation applied to the output, for example shifting an object in an input image will result in an equal shift in its predicted position. Invariance refers to the property that applying some transformation to the function input will not change its output, for example shifting an object in an image should not result in a change of its classification label.

Equivariances to different transformations in deep learning have been studied before. Cohen and Welling [57] generalise convolutional neural networks to group equivariant CNNs (G-CNNs), however for many transformations this may require storing many filters. Gens and Domingos [58] aim to achieve the same goal using Symmetry Networks. Cohen and Welling [59] generalise G-CNNs to steerable CNNs which removes the memory scaling issue and allows working with infinite element groups. Cohen et al. [60] propose gauge equivariant CNNs where the equivariance is to local gauge transformations on the surface of a sphere. Weiler and Cesa [61] use E(2)-equivariant convolutions with steerable CNNs. Henriques and Vedaldi [62] propose warped convolutions which achieve equivariance by warping the input image before passing it through a CNN. Focusing on specific transformations, multiple works deal with equivariance and invariance to rotations [63–71] as well as to scale [72, 73]. In our work we deal with equivariances to continuous transformations (i.e. equivariance to a group with infinite number of elements), but we achieve this by warping the images, unlike for example steerable CNNs [59] which achieves this using kernel properties. The closest work to ours is probably Henriques and Vedaldi [62], however our method is generative while theirs is discriminative, and their method has no guarantees of identifiability.

Equivariant networks have been applied to different areas. For example, Dieleman et al. [66] use rotational invariance for galaxy classification, Han et al. [68] use rotational equivariance for aerial object detection, Lee et al. [74] use equivariance for keypoint detection in images, Pielawski et al. [69] use rotational equivariance for

image registration, Musallam et al. [75] use equivariant features for pose regresssion, and Gupta et al. [70] use rotation equivariance for tracking. While most of the applications of equivariances have been discriminative (i.e. classification, regression), in this work we focus on generative modeling where we use equivariances to generate realistic data never observed at training time (counterfactuals).

2.5 Intuitive Physics

In the intuitive physics literature the aim is to design systems that mimic the human ability to use "common sense" to understand physical phenomena.

Broadly, Duan et al. [76] categorise intuitive physics works by the type of the physical reasoning task each work aims to solve: prediction of the outcome of interactions between physical objects [77–80], prediction of the dynamics (trajectories) of objects [77, 79, 81–83], inference of physical properties of objects [78, 83–88], generating physically plausible future video frames [81, 87–90], predicting whether a scenario is physically possible or not ("violation of expectation") [91, 92], curriculum learning, counterfactual prediction and physical equation prediction [80, 90].

Our work falls most closely in the category dealing with inference of physical properties, as we aim to learn physical variables such as object positions and angles, while also overlapping with the counterfactual prediction and future frame prediction. Unlike existing works, our work is primarily grounded in causality as opposed to the empirical focus in other works, including specialized object detection modules, physics simulators, and deep learning methods which sometimes lack strong theoretical guarantees.

3

Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors

Contents

3.1 Introduction **19**
3.2 Method **21**
3.2.1 Variational autoencoder 22
3.2.2 Additive noise model prior 23
3.2.3 Probabilistic edge penalisation 24
3.2.4 Prediction loss 24
3.3 Experiments **25**
3.4 Conclusion **30**

3.1 Introduction

Humans have an innate ability to reason about the world – if I press the light switch, the light will come on. However, reasoning is notoriously difficult for machines which struggle with more complex reasoning tasks [16]. One reason for this is the fact that most current deep learning methods rely on learning correlations in the data (the so-called IID[1] regime) instead of learning causal relationships [16]. Learning a causal representation of the data would allow machines to reason about what will

[1] Independent and identically distributed.

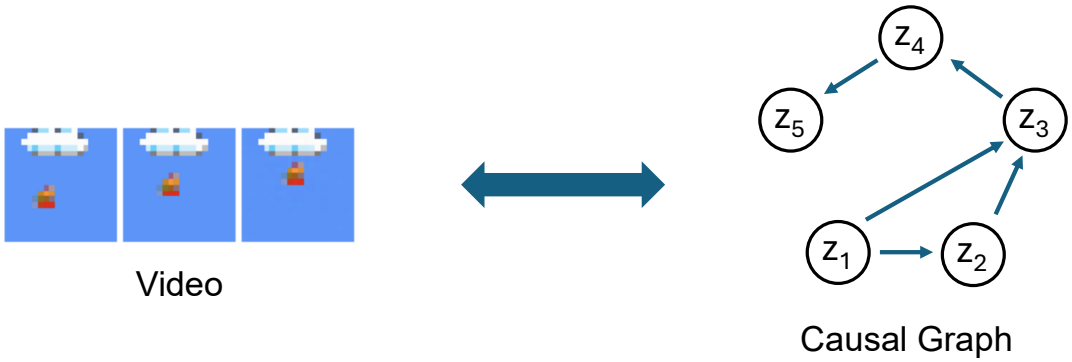


Figure 3.1: The aim of this chapter is to develop a method for learning latent interpretable variables and their causal relationships from short videos.

happen if a certain action is performed (an interventional query) as well as imagine what would have happened if one had acted differently in the past (a counterfactual query) [13]. Therefore, learning causal representations from data is an important problem in current deep learning, especially in the field of reinforcement learning where these representations would allow agents to reason about the world.

The aim of this chapter is to develop a framework for learning interpretable variables and their causal relationships from video data. More specifically, given 3 pairs of frames from video game based data (inspired by Super Mario Bros.) where a character moves at constant speed, we would like to learn a causal graph whose nodes would represent the position of the character at different times and the edges would represent functions relating the character positions over time. This is illustrated in fig. 3.1.

Because of its ability to probabilistically model the data distribution and the interpretability of its latent space, we base our method on one of the standard approaches for representation learning from images, the variational autoencoder (VAE) [36]. In a VAE, an image is encoded into a latent representation, typically an isotropic Gaussian distribution, which is then sampled and decoded back to recover the same image. However, the standard isotropic Gaussian prior encourages learning of independent variables, whereas we would like to learn causally related variables. One of the ways to model the causal relationships between variables is using an additive noise model (ANM), where each variable is a function of the

other variables composed with an additive noise, typically a Gaussian. We thus replace the standard VAE prior with an ANM prior, allowing us to model nonlinear causal relationships between variables (sec. 3.2.2). To ensure computability and ability to model arbitrary causal graphs, we choose a particular ANM where each variable is conditioned on all the previous variables.

The ANM defines a causal graph where each graph node corresponds to a variable and each directed edge corresponds to a function computing the variable from its parents and an additive noise. However, so far this graph can be learned to be arbitrarily complex, whereas it can be argued that most interpretable causal relationships in nature are sparse, only concerning a few variables. To reflect this, we assign a learnable probability to each edge in the causal graph and add a penalty term discouraging learning complex graphs with many edges (sec. 3.2.3). By controlling this penalty term, we are able to learn graphs of different complexities, effectively controlling the balance between graph simplicity and the quality of image reconstruction.

We demonstrate the method on video game based data (sec. 3.3) and show that it recovers the true variables and causal mechanisms up to some transformations. Once the model is learned, we are then able to intervene on the learned causal representation to generate videos successfully extrapolating the system's behaviour into the future, and investigate how the edge penalisation strength affects graph complexity.

3.2 Method

Our method for learning causal graphs from video is based on a Variational Auto-Encoder (VAE) (sec. 3.2.1) with an Additive Noise Model (ANM) prior (sec. 3.2.2) and probabilistic edge penalisation (sec. 3.2.3), with the architecture depicted in fig. 3.2. At a high level, using an ANM prior allows us to learn a broad class of non-linear causal relationships and edge penalisation ensures that the learned graph is sparse, as is often the case in the real world.

Our optimisation objective consists of a reconstruction loss, prediction loss, edge penalisation loss and KL divergence loss. At a high level, the reconstruction loss is

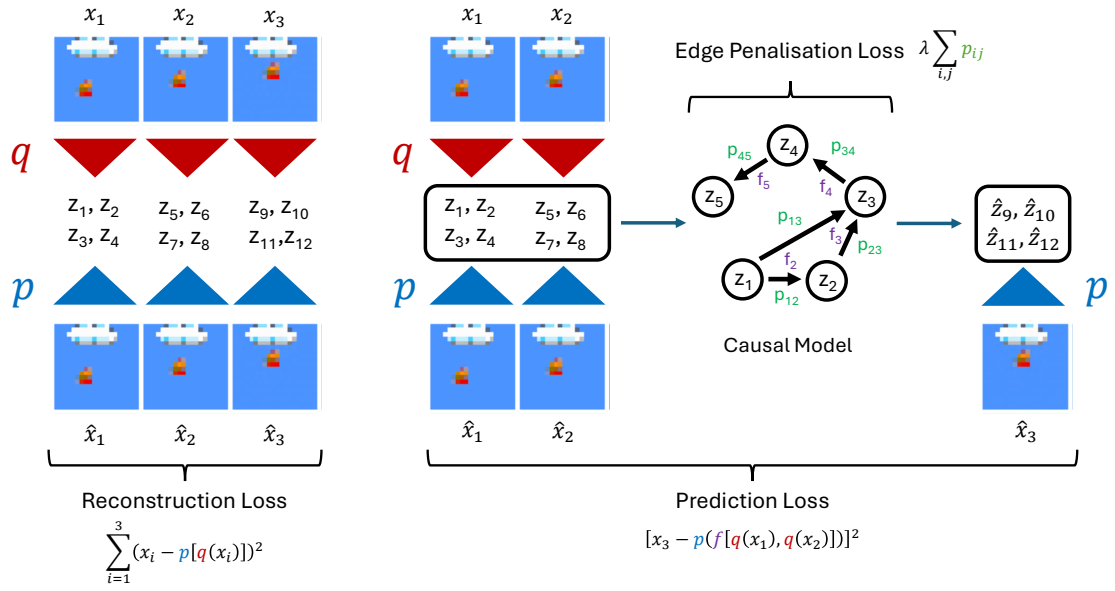


Figure 3.2: Our architecture consists of a Variational Auto-Encoder with an Additive Noise Model prior and probabilistic edge penalisation. The optimisation objective consists of a reconstruction loss, prediction loss, edge penalisation loss and KL divergence loss. See sec. 3.2 for more details.

an L2 loss between each frame x_i and its prediction $\hat{x}_i$ to learn meaningful variables z_i. The prediction loss is an L2 loss between the frame x_3 and its prediction from the causal model, $p(f(q(x_1), q(x_2)))$, to learn meaningful causal mechanisms f_i. The edge penalisation loss penalises the sum of edge probabilities $\sum p_{ij}$ in the causal graph, in order to obtain sparsity. Finally, the KL divergence loss is a loss between the prior and posterior, $D_{KL}(p||q)$. In the following sections we describe each in more detail.

3.2.1 Variational autoencoder

We base our method on an autoencoder, a model that encodes input data into a latent representation and then decodes it back to obtain the same data, with the aim of learning a useful representation of the data. The input data can be an image, and the encoder q and decoder p are typically implemented as deep neural networks (such as convolutional neural networks), with the objective being the minimisation of the distance (such as the L_2 distance) between the input image x and output image $\hat{x} = p(q(x))$, i.e. $\mathcal{L} = L_2(x, \hat{x})$.

To obtain a more structured latent representation with better generative capabilities, we choose to use the variational autoencoder (VAE), that is a type of autoencoder that models the latent bottleneck as a distribution, and its objective is to maximise the likelihood of the data under that distribution given a prior. The optimisation objective for a VAE is given by the evidence lower bound (ELBO) [36],

$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)), \tag{3.1}$$

where $q(z|x)$ is the encoder, $p(x|z)$ is the decoder, $p(z)$ is the prior, and D_{KL} is the Kullback-Leibler divergence. The first term in 3.1 corresponds to the reconstruction loss and the second term to the KL divergence loss (fig. 3.2). Typically, the prior $p(z)$ is chosen to be an isotropic Gaussian to encourage learning independent variables z,

$$p(z) = \mathcal{N}(z|0, \text{diag}(\sigma^2)). \tag{3.2}$$

3.2.2 Additive noise model prior

Because we would like to learn causally related instead of independent variables, we want to replace the isotropic Gaussian prior of a VAE with a different prior where variables depend on each other via (nonlinear) functions. To be able to learn an arbitrary causal graph with N variables, we define our prior to be a fully connected graph $G = (v, e)$ where each variable depends on all the previous variables, i.e.

$$G = (v, e) = (\{z_1, ..., z_N\}, \{z_i \to z_j : i < j, i \leq N, j \leq N\}) \tag{3.3}$$

In this way, any causal graph with N variables can be learned as a special case of our graph (proof in appendix A.1). To be able to learn nonlinear relationships between variables, we parametrise the graph edges by an additive noise model, defined as

$$z_i = f_i(z_1, ..., z_{i-1}) + n_i, \quad i \leq N \tag{3.4}$$

where f_i is the mechanism for obtaining z_i from its parents and n_i is an additive noise. For simplicity we choose n_i to be Gaussian noise with zero mean and variance σ_i^2 and thus obtain the prior

$$p(z) = \prod_i^N \mathcal{N}(z_i|f_i(z_1, ..., z_{i-1}), \sigma_i^2) \tag{3.5}$$

The prior in eq. 3.5 defines a causal graph with nodes z_i and edges f_i (fig. 3.2, causal graph). We can now use this prior with the loss in eq. 3.1 to learn the causal variables $z_1, ..., z_N$ and the mechanisms $f_1, ..., f_N$ from data.

3.2.3 Probabilistic edge penalisation

Because it can be argued that most interpretable causal relationships in nature are sparse, we would like to modify our fully connected graph G to allow for modeling the presence or absence of its edges. We do this by defining $p_{i,j}$ as the probability of existence of the edge $z_i \to z_j$. Because graph edges are either present or absent (a binary variable), this makes the learning difficult. To solve this, we parametrise $p_{i,j}$ using a binary Gumbel-Softmax distribution [93] that allows us to learn the edge probabilities in a differentiable way.

To encourage sparsity of the learned graphs, we take inspiration from Nowzohour and Bühlmann [94], who prove that penalising the maximum likelihood score of different graphs by a factor proportional to the number of graph edges allows one to obtain the true causal graph as the one with the maximum score. We thus augment our loss in eq. 3.1 by an edge penalisation term,

$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z)) - \lambda|G| \tag{3.6}$$

where $|G|$ denotes the number of edges of the graph G and λ denotes the penalisation strength. We approximate $|G|$ in eq. 3.6 with our probabilistic edge model as

$$|G| = \sum_{i,j} p_{i,j} \tag{3.7}$$

which reduces to the number of graph edges when $p_{i,j}$ is binary (fig. 3.2, edge penalisation loss). Therefore, we obtain the optimisation objective in eq. 3.6 with the prior $p(z)$ defined in eq. 3.5 and the number of edges $|G|$ defined in eq. 3.7.

3.2.4 Prediction loss

While our optimisation objective contains a prior defining a causal graph (eq. 3.5), in its current form the causal mechanisms f_i can be learned to be arbitrary. To constrain these, we augment the optimisation objective in eq. 3.6 with a prediction loss term,

$$\mathbb{E}_{q(z|x_1,x_2)}[\log p(x_3|f(z))] \tag{3.8}$$

which corresponds to the L2 loss from encoding the first two frames x_1 and x_2 into latent variables z, passing them through the causal model f, and decoding the result to obtain the prediction for the third frame $\hat{x}_3$ which is compared with the actual frame x_3 (fig. 3.2, prediction loss). This formulation ensures that the functions f_i will correspond to temporal relationships between the latent variables z from the first two frames (x_1, x_2) and the variables from the third frame (x_3). The final optimisation objective is thus given by

$$\mathcal{L} = \mathbb{E}_{q(z|x)}[\log p(x|z)] + \mathbb{E}_{q(z|x_1,x_2)}[\log p(x_3|f(z))] - D_{KL}(q(z|x)||p(z)) - \lambda|G| \tag{3.9}$$

3.3 Experiments

In this section we validate our method experimentally. We use video game based data with a game character moving at a constant speed and show that our method recovers the positions of the character in different frames of the video and their causal relationships over time up to scales and rotations, using a learnable linear prior.

We first create a dataset based on the Super Mario Bros video game depicting an object moving with linear motion in different directions, at different speeds, and on varied backgrounds. Each video consists of three frames, with the object's coordinates (x_i, y_i) for $i \in \{1, 2\}$ sampled uniformly from $[7, 12]$, and the third obeying $x_3 = 2x_2 - x_1, y_3 = 2y_2 - y_1$, in order to make the object move linearly. We then constrain the prior so that each video frame corresponds to 4 latent variables in the causal graph, where each variable z_i causally depends on all z_j where $j < i$. We also constrain the causal mechanisms f_i to be linear, as we wish to learn linear motion from the video. We use the architecture in fig. 3.2 (and append an encoded background frame to the the decoder's input to allow the latent variables to ignore the background) and optimise the objective in eq. 3.9, selecting the sparsest graph that still achieves the best reconstruction accuracy.

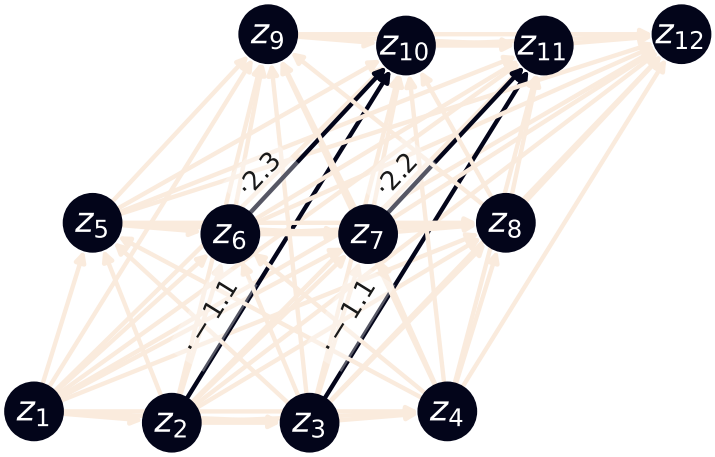


Figure 3.3: Learned graph edges and mechanisms relating the latent variables, recovering the data generation process up to a scaling factor.

Learning a causal graph. Figure 3.3 shows the graph learned by the causal-prior VAE, with the bottom row (z_1, z_2, z_3, z_4) corresponding to latent variables at time $t = 1$, the middle row (z_5, z_6, z_7, z_8) corresponding to $t = 2$ and the top row $(z_9, z_{10}, z_{11}, z_{12})$ corresponding to $t = 3$. Learned edges are shown in black and missing ones in beige; learned functions f_i are represented as a number (multiplicative factor) next to the corresponding edge, since they are linear in this case. The graph shows that each component of the object's 2D position at time $t = 3$ (expressed as (z_{10}, z_{11})) depends on its 2D positions at times $t = 1$ (z_2, z_3) and $t = 2$ (z_6, z_7), following the model $z_{10} = f_{10}(z) = 2.3z_6 - 1.1z_2$ and $z_{11} = f_{11}(z) = 2.2z_7 - 1.1z_3$, which matches the data generation process up to a scale factor.

Latent response to position. Figure 3.4 shows the response of each latent variable as the object's horizontal and vertical position is varied on the input. The rows correspond to variables from times $t = 3, 2, 1$ respectively. We can see that z_2, z_6, z_{10} encode the object's position along the bottom-right to top-left diagonal at different moments in time, while z_3, z_7, z_{11} encode the position along the orthogonal direction. The remaining variables do not exhibit any significant response to position. This shows the variables correctly match the data generation process up to a rotation.

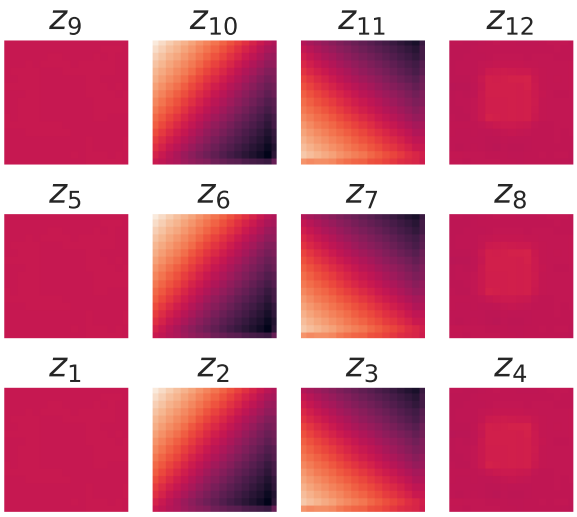


Figure 3.4: Latent response of variables to the object's horizontal and vertical position, recovering the data generation process up to a rotation.

Interventions visualisation. Once a causal model relating the latent variables has been learned, we can perform interventions on some of its variables. In Figure 3.5 we encode a randomly sampled reference clip from the dataset and separately intervene by assigning different values (rows) to variables $z_1, ..., z_{12}$ (columns), propagating the changes to their child variables using the SEM, and decoding the results. The results are decoded using a black background and the resulting 3-frame decoded sample is combined into a single frame by colouring the $t = 1$ decoded image by red, the $t = 2$ image by white, and $t = 3$ by green. We observe that intervening on the variables for $t = 1$ (z_2, z_3) changes both the object's position in $t = 1$ (red, columns 1 and 2) and $t = 3$ (green, columns 1 and 2), following the SEM. Intervening on the variables for $t = 2$ (z_6, z_7) changes the object's $t = 2$ position (white, columns 3 and 4) and $t = 3$ position (green, columns 3 and 4); while intervening on the variables for $t = 3$ (z_{10}, z_{11}) changes only its $t = 3$ position (green, columns 5 and 6), which reflects the fact that other positions (in the past) do not depend on it. Intervening on other variables has no effect (last column). This confirms that the model has learned the correct temporal causal structure, as each intervention affects the right variables.

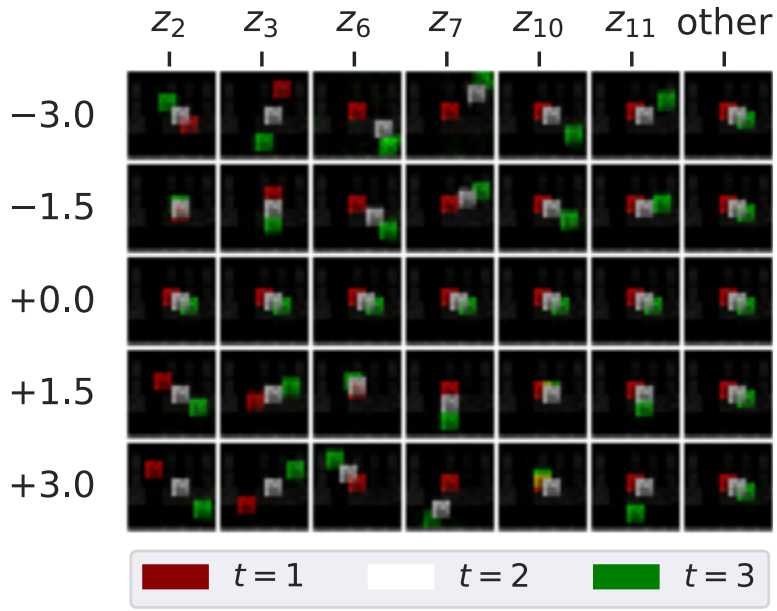


Figure 3.5: Intervention results showing that intervening on each variable propagates to its descendants through the correct mechanism.

Extrapolation visualisation. Having learned a SEM relating variables at time t to the variables at times $(t-1, t-2)$ now makes it possible to extrapolate the variable values into the future. In Figure 3.6 we start with 3 samples (rows) of 2 consecutive frames from the dataset x_1, x_2, encode them to obtain the $t = 1, 2$ latent variables $(z_1, ..., z_8)$, and iteratively pass them through the SEM to compute the $t = 3$ latent variables $(z_9, ..., z_{12})$, decoding these into $\hat{x}_3$. We now repeat this process by taking $z_5, ..., z_{12}$ as the $t = 1, 2$ variables and use the SEM to predict the $t = 3$ variables which we decode into $\hat{x}_4$ and similarly for $\hat{x}_5$, thus obtaining predictions for 3 time steps into the future. The future predictions confirm the knowledge learned in the SEM that the object moves linearly and can be used to predict future frames accurately.

Effect of edge penalisation on reconstruction accuracy. Figure 3.7 shows the reconstruction accuracy achieved with the model as a function of the parameter controlling graph sparsity (λ in eq. 3.9) after training for 250 epochs. The plot shows that for weak edge penalisation the reconstruction accuracy is good (around

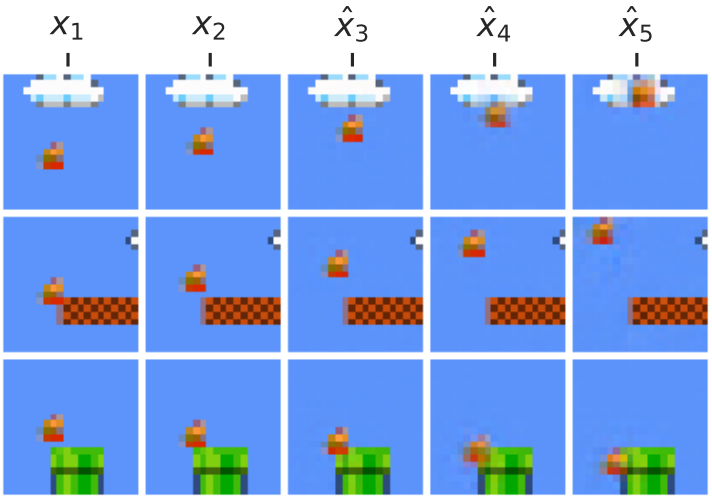


Figure 3.6: Extrapolation results obtained by iteratively passing the encoded variables through the learned SEM to predict their future values.

99.7%) while if the edge penalisation is too large the accuracy drops (to around 99.1%). This is because for weak edge penalisation the graph is relatively dense which allows the SEM to model the time-based causal relationships between the variables and when the penalisation is too large the graph becomes too sparse to be able to model these relationships. Somewhere around the value of the coefficient 10^{-2} the graph becomes as sparse as possible while still keeping the reconstruction accuracy high, and this is the area from which we select the graph.

Effect of edge penalisation on graph correctness. Figure 3.8 shows the Structural Hamming Distance (SHD) between the learned and the ground truth graph as a function of the parameter controlling graph sparsity (λ in eq. 3.9) after training for 250 epochs. The SHD is computed by counting how many edges need to be inserted or removed to obtain the ground truth graph (up to a permutation of variables within each time step). When the coefficient is below around 10^{-3}, the edge penalisation is too weak, resulting in a graph with too many edges (thus a high SHD). In contrast, for the coefficient above around 10^{-2}, the graph becomes too sparse with no edges (thus also resulting in high SHD). The intermediate region

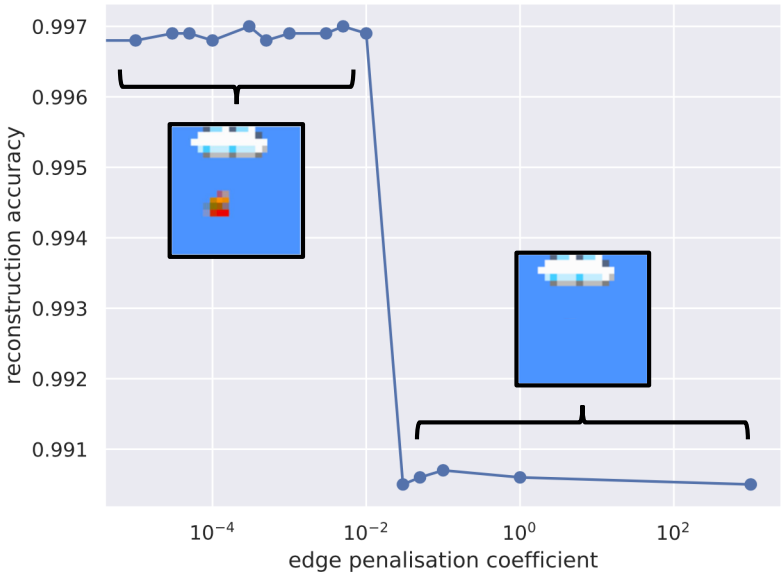


Figure 3.7: Reconstruction accuracy as a function of parameter controlling graph sparsity. For low edge penalisation, the graph is dense enough to obtain good reconstruction accuracy; for high edge penalisation, the graph is too sparse, resulting in low reconstruction accuracy.

between 10^{-3} and 10^{-2} corresponds to the region where the learned graph has exactly the same structure as the ground truth graph (and thus SHD is zero).

3.4 Conclusion

In this work we developed a model that extends the Variational Auto-Encoder framework with an Additive Noise Model prior and edge sparsity penalisation that allows one to learn causal variables and mechanisms from video data. By adjusting the prior and the penalisation strength, the model is able to learn graphs with different levels of complexity. We performed experiments with video game based data, where the model learned the true causal graph corresponding to object positions and their temporal relationships up to rotations and scalings. Once the graph was learned, we were then able to intervene on its variables to generate videos visualising hypothetical scenarios and predicting the future behaviour of the system.

While general, this method may become underconstrained unless one imposes strong priors on the form of the causal mechanisms and the variables. Aiming to obtain stronger identifiability guarantees, in the next chapter we will restrict our

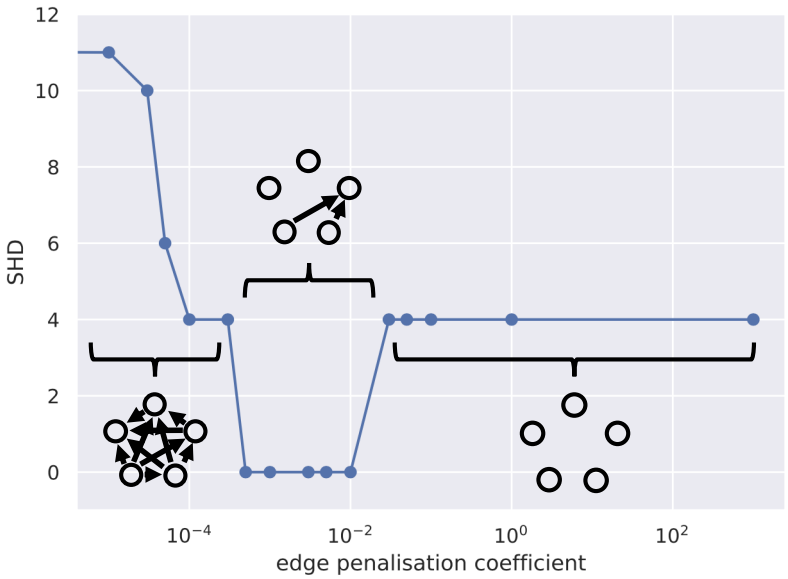


Figure 3.8: Structural Hamming Distance (SHD) as a function of parameter controlling graph sparsity. For low edge penalisation, the SHD is high because the graph is too dense; for high edge penalisation, the SHD is high because graph is too sparse; for intermediate values of edge penalisation, the SHD is low because the learned graph corresponds to the true graph.

focus to a system with a single variable that is a function of time, and show its successful application on a range of real world physics experiment videos.

4

Interpretable Representation Learning from Videos using Nonlinear Priors

Contents

4.1 Introduction **33**
4.2 Method **35**
4.2.1 Optimisation Objective 36
4.2.2 Posterior GMM 36
4.2.3 Prior GMM 36
4.2.4 KL Divergence 38
4.2.5 Multivariate case 39
4.3 Experiments **40**
4.3.1 Implementation 41
4.3.2 Mass on a Spring 42
4.3.3 Pendulum 43
4.3.4 Falling Object 45
4.3.5 Pulsar 46
4.4 Discussion **47**
4.5 Conclusion **48**

4.1 Introduction

Humans have an innate ability to observe the world and form a mental model of how things work – if I release a hammer it will fall to the ground. Even more fascinating is the human ability to also reason about hypothetical scenarios which

they may never have observed before – what would happen if I dropped the same hammer while standing on Jupiter instead of on the Earth? In this example having a good mental model of how the position of a hammer depends on gravitational acceleration enables one to reason about the hammer's position on planets with different gravity, even though one has never actually performed such experiment there. In the causality literature this is referred to as counterfactual reasoning [13, 95], and it relies on having a Structural Equation Model describing the data that can be intervened on to obtain answers to counterfactuals, i.e. hypothetical questions.

There is a vast literature on traditional causality where many works aim to identify causal relations between variables that are provided by domain experts (such as economists or medical personnel) [27, 96], however in many real world cases the variables are not specified a priori – instead we want to learn them. Separately, in the field of intuitive physics researchers have been working on developing machine learning models that can understand physical scenarios [77–92], often with the aim to predict future behaviour of a system – however, the works seldom focus on answering counterfactual questions in different situations.

In between the two, the field of causal representation learning [16] has recently emerged to train models to predict variables that are causally related, which can include physical variables. This requires somewhat strong assumptions, as it has been shown that learning a causal model purely from observational data is not possible [43]. Some works rely on the knowledge of interventions to infer the correct causal structure [97], while other works make assumptions on the distribution of the latent variables and the causal mechanisms [98–104]. Instead, in our work we assume a single variable and a known causal mechanism that is an additive noise model, which allows us to guarantee that the learned representation is identifiable and interpretable.

More specifically, if we consider the case of a hammer being dropped to the ground, we know that its vertical position depends quadratically on time and linearly on the gravitational acceleration ($y = \frac{1}{2}gt^2$, Newton's second law), which enables us to learn a variable corresponding to the object's position (assuming there are no

confounders). We can then intervene on this model by changing the gravitational acceleration to reason about the object falling on different planets and visualise the resulting videos. In this chapter we learn such models for different physical scenarios, including a falling object, an object oscillating on a spring, and object swinging on a pendulum, and a pulsar oscillating in the Crab nebula. For each video of a real-world physics experiment we specify a prior (physical equation) that can be intervened to generate an imagined, previously-unobserved case of the same scenario but with different physical properties, specifically an object falling on Jupiter, a pendulum with added air resistance, a mass on a spring oscillating with a different mass and frequency, and a pulsar pulsating at a different frequency.

To learn the physical variables we use a Variational Auto-Encoder (VAE) [36]. However, instead of a standard isotropic Gaussian prior, we use an Additive Noise Model (ANM) which is a general model of the form $y = f(x)+n$ (where n is Gaussian noise), that can describe a wide range of physical systems – for example, for a video of a mass oscillating on a spring, we can specify the prior as $y(t) = A\cos(\omega t) + n$. Note that we assume a single variable and a known physical mechanism, which is crucial to obtain identifiability. We then devise a method to approximate this prior as a Gaussian Mixture Model (GMM) and derive an expression for the KL divergence estimate to be able to optimise the data log-likelihood of the VAE. Once we train the model using a given ANM prior we can then intervene on the prior and decode a video corresponding to an imagined scenario given by the new model – for example, $y(t) = \frac{1}{2}A\cos(2\omega t) - \frac{1}{2} + n$. In this way we have developed a method for counterfactual reasoning in videos.

4.2 Method

We will now describe our main proposal, which extends VAEs with a causal prior for temporal data.

4.2.1 Optimisation Objective

We start from a standard VAE [36], and consider the case when all input samples correspond to different temporal samples from the same video. The optimisation objective is to maximise the evidence lower bound [36]:

$$\text{ELBO} = \mathbb{E}_{p(x)}[\mathbb{E}_{q(z|x)} \ln p(x|z) - D_{KL}(q(z|x)||p(z))] \tag{4.1}$$

The first term in eq. 4.1 corresponds to a reconstruction error, given by the L_2 loss between the original image x_t and its reconstruction $\hat{x}_t$. The second term in eq. 4.1 corresponds to the Kullback-Leibler (KL) divergence between the posterior and the prior distributions. We compute each as follows.

4.2.2 Posterior GMM

The posterior distribution consists of a mixture of T Gaussians, each corresponding to a single frame of video (sample). Each frame x_t is encoded independently by an encoder q to obtain the parameters μ_{y_t} and σ_{y_t} which are concatenated with time t to obtain the posterior Gaussian

$$\mathcal{N}(\mu_t^q, \Sigma_t^q) = \mathcal{N}\left(\begin{bmatrix} t \\ \mu_{y_t} \end{bmatrix}, \begin{bmatrix} \epsilon^2 & 0 \\ 0 & \sigma_{y_t}^2 \end{bmatrix}\right) \tag{4.2}$$

In all experiments we set $\epsilon = 0.01$. See fig. 4.1 (top branch of the bottom panel) for an illustration. To compute the reconstruction loss (fig. 4.1, top panel) we only pass the second component of the sampled posterior to the decoder, namely $y_t \sim \mathcal{N}(\mu_{y_t}, \sigma_{y_t}^2)$, while discarding the first component $t \sim \mathcal{N}(t, \epsilon^2)$, in order to prevent the decoder solely relying on the time variable and ignoring the y_t variable.

4.2.3 Prior GMM

First, we specify the prior by choosing the time distribution $p_T(t)$, the noise distribution $p_N(n)$, and the mechanism $y = f(t) + n$, based on our knowledge of the physical process in a given video (which must be given). The idea is then to sample enough time instants $t_0 \sim p_T(t_0)$ and noise realizations $n_0 \sim p_N(n_0)$ to construct a Gaussian mixture model from the prior, where each Gaussian has

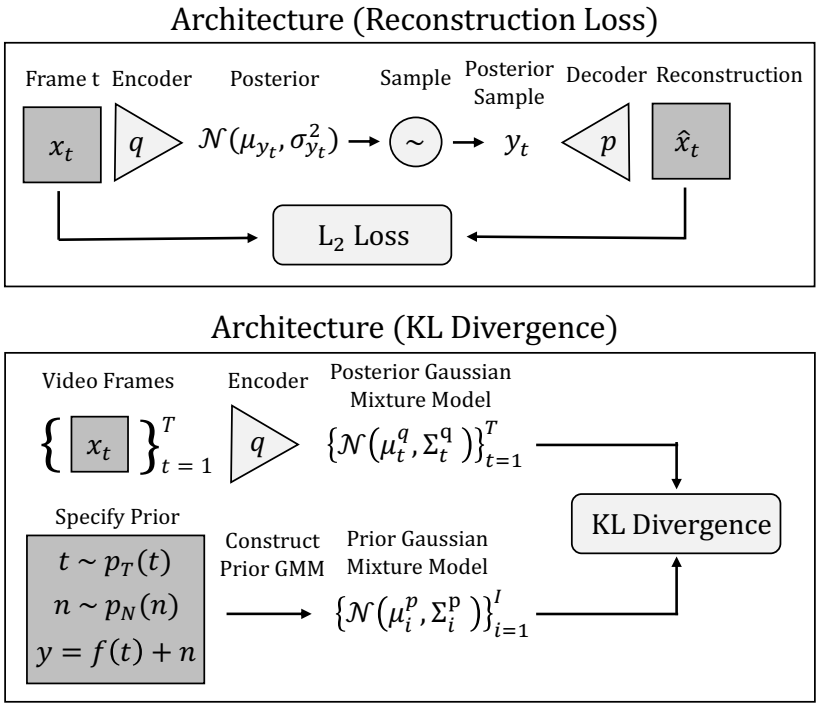


Figure 4.1: Top: architecture to compute the reconstruction loss. Bottom: architecture to compute the KL divergence loss. See sec. 4.2 for details.

mean (t, y) with $t = t_0$ and $y = f(t_0) + n_0$, and with a covariance that follows the curvature of $f(t)$ at t_0.

Formally, we model each time as a Gaussian[1] $t \sim \mathcal{N}(t_0, \sigma_t^2)$ centered around $t_0 \sim p_T(t_0)$ (e.g. $t_0 \sim U(0, 1)$) with variance σ_t^2 set to a fixed value (here $\sigma_t = \epsilon = 0.01$). Next, because a non-linear function of a Gaussian variable is no longer Gaussian, we approximate $y = f(t) + n$ linearly around t_0 using its first order Taylor expansion as

$$y \approx f(t_0) + \left.\frac{\partial f}{\partial t}\right|_{t_0} (t - t_0) + n. \tag{4.3}$$

Using eq. 4.3 the joint distribution of t and y can now be written as

$$\begin{bmatrix} t \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ \left.\frac{\partial f}{\partial t}\right|_{t_0} \end{bmatrix} t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} n + \begin{bmatrix} 0 \\ f(t_0) - \left.\frac{\partial f}{\partial t}\right|_{t_0} t_0 \end{bmatrix}. \tag{4.4}$$

[1]The time t is modelled as a Gaussian whose mean is set to the true time of the corresponding image frame and whose variance models the uncertainty in temporal sampling, which is related to the time difference between two consecutive image frames.

Using the formula for linear combinations of Gaussian variables ([105] Sec. 8.1.4) the joint probability in eq. 4.4 can be expressed as $(t, y) \sim \mathcal{N}(\mu^p, \Sigma^p)$ where

$$\mu^p = \begin{bmatrix} 1 \\ \frac{\partial f}{\partial t}\Big|_{t_0} \end{bmatrix} t_0 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} n_0 + \begin{bmatrix} 0 \\ f(t_0) - \frac{\partial f}{\partial t}\Big|_{t_0} t_0 \end{bmatrix} \tag{4.5}$$

$$\Sigma^p = \begin{bmatrix} 1 \\ \frac{\partial f}{\partial t}\Big|_{t_0} \end{bmatrix} \sigma_t^2 \begin{bmatrix} 1 \\ \frac{\partial f}{\partial t}\Big|_{t_0} \end{bmatrix}^T + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \sigma_n^2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T, \tag{4.6}$$

where we have used $t \sim \mathcal{N}(t_0, \sigma_t^2)$ and $n \sim \mathcal{N}(n_0, \sigma_n^2)$. Finally, simplifying equations 4.5 and 4.6 yields the prior

$$\begin{bmatrix} t \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} t_0 \\ f(t_0) + n_0 \end{bmatrix}, \begin{bmatrix} \sigma_t^2 & \frac{\partial f}{\partial t}\Big|_{t_0} \sigma_t^2 \\ \frac{\partial f}{\partial t}\Big|_{t_0} \sigma_t^2 & \left(\frac{\partial f}{\partial t}\Big|_{t_0}\right)^2 \sigma_t^2 + \sigma_n^2 \end{bmatrix} \right). \tag{4.7}$$

The prior GMM is thus constructed as a mixture of the Gaussians specified in eq. 4.7 with equal weight. See fig. 4.1 (bottom branch of the bottom panel) for illustration.

4.2.4 KL Divergence

Because the prior and posterior are both GMMs (as opposed to single Gaussians as in a standard VAE), the KL divergence can no longer be computed in closed form. We can nevertheless approximate the KL divergence using Monte Carlo sampling [105] (chapter 11) as

$$D_{KL}(q||p) \approx \frac{1}{I} \sum_{i=1}^{I} [\ln q(x_i) - \ln p(x_i)], \;\; x_i \sim q(x) \tag{4.8}$$

In words, we first sample the posterior GMM to obtain x_i, and then we evaluate this sample in the GMM prior $p(x)$ and posterior $q(x)$ to obtain the estimate, and we repeat this I times. However, naively evaluating p and q in eq. 4.8 introduces many numerical instabilities, and the expression is not immediately parallelizable over both the posterior samples (x_i) and the prior and posterior Gaussian mixtures (p, q) at the same time. To address both issues we derive a numerically stable and parallelizable form of eq. 4.8, by deriving the form of $\ln p(x_i)$ and $\ln q(x_i)$ directly in terms of the parameters of individual Gaussians in each

GMM. Substituting the form of a GMM for $p(x)$ and evaluating the probability density function (PDF) of each Gaussian yields

$$\begin{aligned}\ln p(x_i) &= \ln \sum_{j=1}^{N} \frac{1}{N} \mathcal{N}(x_i; \mu_j, \Sigma_j) \\ &= \ln \sum_{j=1}^{N} \frac{1}{N} (2\pi)^{-\frac{k}{2}} \det(\Sigma_j)^{-\frac{1}{2}} \exp\Big[-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)\Big]. \end{aligned} \tag{4.9}$$

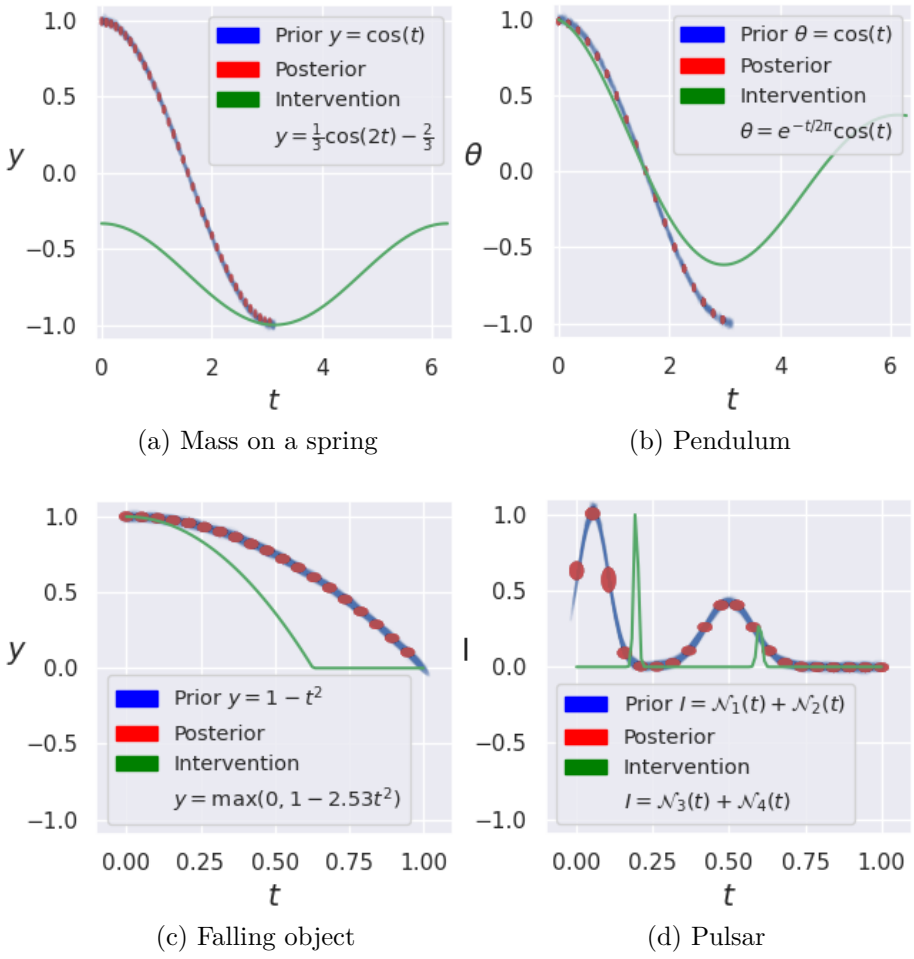
Simplifying eq. 4.9 and factoring terms to merge numerically unstable operations (such as independently-computed determinants and logarithms) yields

$$\ln p(x_i) = -\ln N - \frac{k}{2}\ln 2\pi + \overbrace{\ln \sum_{j=1}^{N} \exp}^{\texttt{logsumexp}}\Big[-\frac{1}{2}\underbrace{\ln(\det(\Sigma_j))}_{\texttt{logdet}} - \frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)\Big]. \tag{4.10}$$

Eq. 4.10 now only contains numerically stable operations, namely linear combination of terms, `logsumexp` and `logdet`. It can also be efficiently computed for an arbitrary number of posterior samples (indexed by i) and prior GMM Gaussians (indexed by j) in parallel, unlike eq. 4.8. An identical expression to eq. 4.10 can be derived for the posterior $q(x_i)$ instead of the prior $p(x_i)$ and subtracting the two according to eq. 4.8 yields the KL divergence estimate.

4.2.5 Multivariate case

In this chapter we explore the case of a single variable y. However, the method can be extended to a multivariate case $y = [y_1, ..., y_n]$ where each y_i can be learned using the proposed method by intervening on the video such that only the part of the video corresponding to y_i is allowed to vary while all other parts corresponding to $y_{\setminus i}$ are fixed. For example, for a video with two masses on a spring, one might first learn y_1 by perturbing the first pendulum and then learn y_2 by perturbing the second pendulum, and then vary y_1, y_2 jointly to generate videos of the two masses moving simultaneously.



(a) Mass on a spring (b) Pendulum

(c) Falling object (d) Pulsar

Figure 4.2: Latent space visualisation showing the prior generated using the linearization method (blue), posterior generated by encoding many images from the dataset (red), and the intervened prior (green), shown for experiments (a-d).

4.3 Experiments

In this section we demonstrate our method on 4 physics experiments: a mass oscillating on a spring, a mass swinging on a pendulum, an object falling from a height and a pulsar pulsating over time. Even though our method only deals with a single variable with a known mechanism, these experiments demonstrate its practical applications. Each experiment has high-dimensional observations, in the form of a raw RGB video. For each experiment we specify a physics prior for the variable that we want the model to learn, and once the model is learned the prior is intervened on to generate a never-before-seen video of a modified experiment – for example seeing how a pendulum would behave with air resistance or seeing how an object

Experiment	Time Prior	Mechanism Prior	Mechanism Intervention
Spring	$t \sim U(0, 2\pi)$	$y = \cos(t)$	$y = \frac{1}{3}\cos(2t) - \frac{2}{3}$
Pendulum	$t \sim U(0, 2\pi)$	$\theta = \cos(t)$	$\theta = e^{-t/2\pi}\cos(t)$
Fall	$t \sim U(0, 1)$	$y = 1 - t^2$	$y = \max(0, 1 - 2.53t^2)$
Pulsar	$t \sim U(0, 1)$	$I = 0.125\ \mathcal{N}(t; 0.055, 0.05^2)$ $+0.085\ \mathcal{N}(t; 0.5, 0.08^2)$	$I = 0.0193\ \mathcal{N}(t; 0.195, 0.007^2)$ $+0.006\ \mathcal{N}(t; 0.6, 0.08^2)$

Table 4.1: Summary of the prior time distribution, the prior mechanism equation and the intervened mechanism equation for the experiments with mass on a spring, pendulum, falling object, and Crab pulsar.

	Standard VAE		Temporal VAE		Proposed Method	
	Lat. MSE	Rec. Acc.	Lat. MSE	Rec. Acc.	Lat. MSE	Rec. Acc.
Spring	1.96	99.41%	2.00	**99.95%**	$\mathbf{2.7 \cdot 10^{-4}}$	**99.95%**
Pendulum	1.99	99.69%	1.98	99.95%	$\mathbf{2.5 \cdot 10^{-4}}$	**99.99%**
Fall	2.03	98.33%	2.00	**99.96%**	$\mathbf{2.0 \cdot 10^{-3}}$	99.85%
Pulsar	1.99	95.82%	1.97	**99.98%**	$\mathbf{7.0 \cdot 10^{-2}}$	97.56%

Table 4.2: Mean squared error (Lat. MSE, lower is better) of the predicted posterior latent variables w.r.t. estimated ground truth physical variables, and reconstruction accuracy (Rec. Acc., higher is better) of the decoded video frames w.r.t. input video frames. Results are shown for the proposed method, a standard VAE and a temporally conditioned VAE, for each experiment.

would fall on Jupiter instead of on Earth. This method enables one to visualise many hypothetical scenarios (counterfactuals) that have never been seen before at training time but still result in realistic, physically plausible videos. The equations of the prior and interventions are summarised in table 4.1 and are visualised in fig. 4.2, and the corresponding prior and intervened videos are shown in fig. 4.3.

4.3.1 Implementation

We implement the method described in sec. 4.2 using the architecture in fig. 4.1. The encoder q and decoder p are both 4-layer multilayer perceptrons (MLPs) with 128 hidden units and with ReLU activations between each layer. We optimise the ELBO term (eq. 4.1) using the Adam optimiser [106] with learning rate $3 \cdot 10^{-4}$ and batch size 100, until convergence. Image dimensions are $3 \times 100 \times 226$ px for the Spring experiment, $3 \times 100 \times 108$ px for the Pendulum experiment, $3 \times 100 \times 205$ px for the Falling Object experiment, and $3 \times 77 \times 78$ px for the Pulsar experiment.

Baselines. In each experiment we train two baseline VAEs with a standard $\mathcal{N}(0, 1)$ prior [36]. The decoder of one baseline has the video time appended to its input, while the other does not (Temporal VAE and Standard VAE in table 4.2). All other training details are identical to our model.

4.3.2 Mass on a Spring

In this video (fig. 4.3a, top) a mass is hung from a spring and is left to oscillate with a certain oscillation amplitude and frequency around a certain height, with negligible air resistance. Assuming mass m, spring constant k, acceleration due to gravity g and vertical mass position y, the equation of motion for the mass can be derived by solving Newton's second law for y:

$$m\frac{\partial^2 y}{\partial t^2} = -ky + mg \tag{4.11}$$

Assuming the mass starts at its highest position, the solution to eq. 4.11 is given by

$$y(t) = A\cos(\omega t) + h, \quad \omega = \sqrt{\frac{k}{m}}, \; h = \frac{mg}{k} \tag{4.12}$$

For simplicity we set $A = 1$, $\omega = 1$ and $h = 0$ in eq. 4.12 to obtain the prior $y(t) = \cos(t)$.

Quantitative results. We plot the prior (blue) and posterior (red) in fig. 4.2a and note that the posterior matches the prior very closely. In table 4.2 (first row) we show that our method achieves the lowest mean squared error (MSE) between the predicted and the estimated ground truth position of the object (after normalising the scale and offset of each distribution) among all baselines by 3 to 4 orders of magnitude, showing that our model has learned the correct physical variables. Finally, in table 4.2 (first row) we show that our model achieves a similarly good reconstruction accuracy (Rec. Acc.) as the temporal VAE baseline while outperforming the standard VAE, where the (seemingly small) 0.5% difference in accuracy means that the standard VAE only reconstructed the average frame across the whole video, while our method and the temporal VAE correctly reconstructed each video frame.

Intervention results. As an example of counterfactual reasoning, consider that we want to generate a video of an experiment where the amplitude of oscillation is a third of the original one ($A_{int} = 1/3A$), the mass is four times lighter ($m_{int} = m/4 \implies \omega_{int} = 2\omega$) and the oscillation point is two thirds below the original one ($h_{int} = h - 2/3$). Thus, for such an intervention we can modify the prior as $y_{int}(t) = 1/3\cos(t) - 2/3$ (fig. 4.2a, green) and decode it to obtain the corresponding video (fig. 4.3a, bottom). The counterfactual video shows a realistic demonstration of a mass oscillating around a lower point with a smaller amplitude and a higher frequency.

4.3.3 Pendulum

In this video (fig. 4.3b, left) a mass is suspended from a string and is left to oscillate from left to right to form a pendulum with negligible air resistance. Assuming mass m, string length L, string angle from vertical θ, coefficient c related to air resistance (for generality) and acceleration due to gravity g, the equation of motion for the mass can be derived by solving Newton's second law:

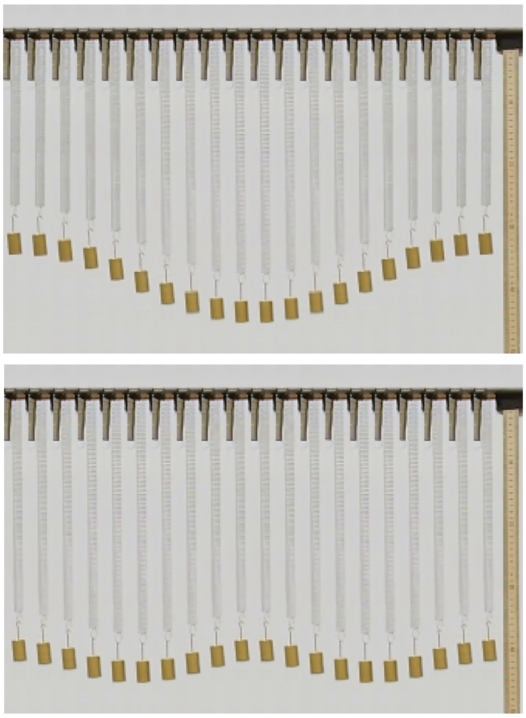
$$mL\frac{\partial^2\theta}{\partial t^2} = -mg\sin(\theta) + cL\frac{\partial\theta}{\partial t} \tag{4.13}$$

If the mass is released from angle θ_0 eq. 4.13 is solved by

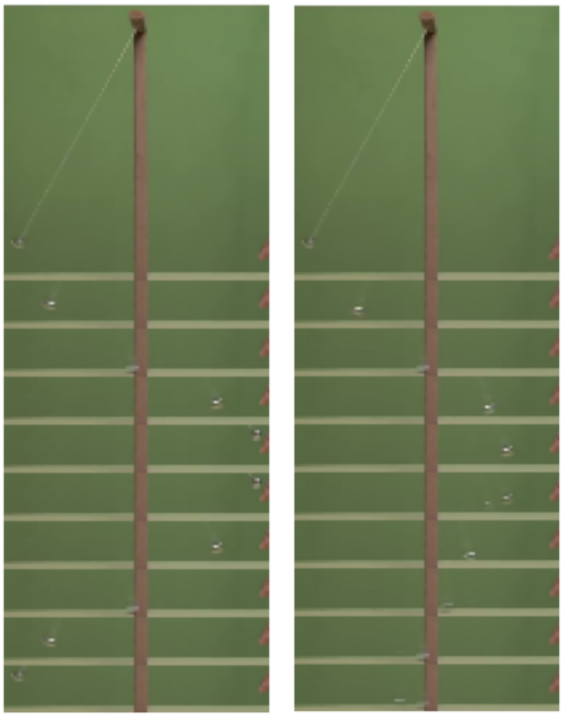
$$\theta(t) = \theta_0 e^{-\gamma t}\cos(\omega t), \quad \omega = \sqrt{\frac{g}{L}},\ \gamma = \frac{c}{2m} \tag{4.14}$$

For simplicity we set $\theta_0 = 1$, $\omega = 1$ and assume no air drag by setting $\gamma = 0$ in eq. 4.14 to obtain the prior $\theta(t) = \cos(t)$.

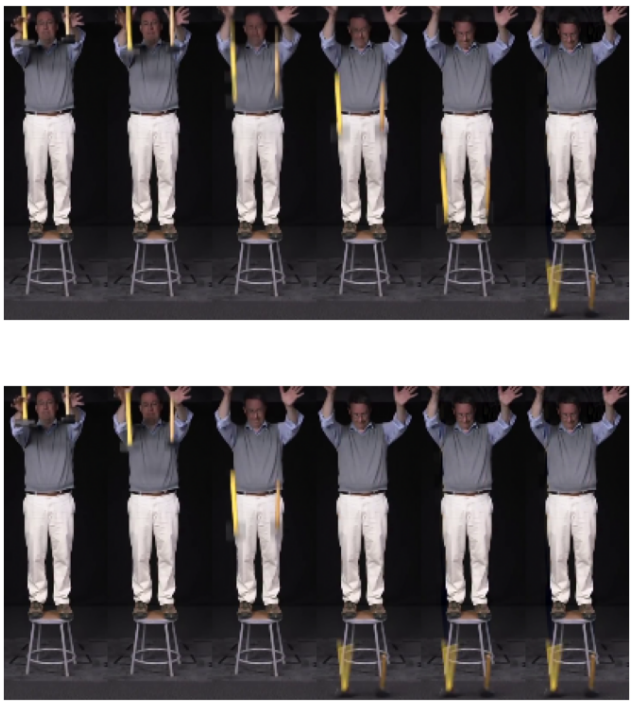
Quantitative results. We plot the prior (blue) and posterior (red) in fig. 4.2b and note the posterior matches the prior very closely. Similar to the spring experiment, table 4.2 (second row) shows that our method achieves the lowest MSE among all baselines and a reconstruction accuracy similarly good to the temporal VAE, demonstrating that our method has learned the correct physical variable (pendulum angle) while successfully reconstructing the images unlike the baselines.



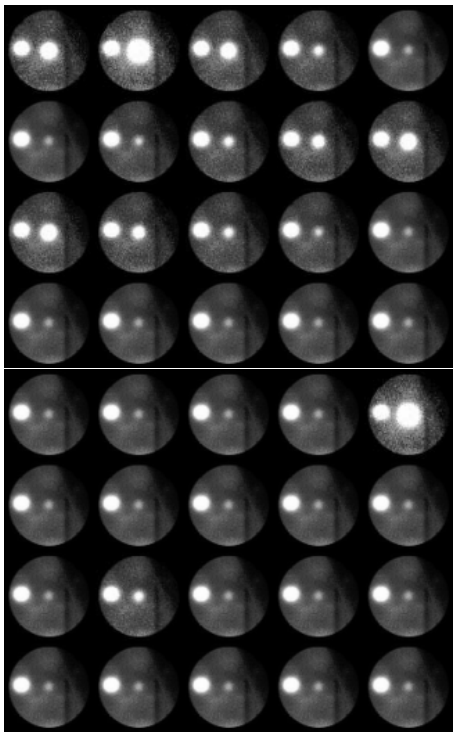
(a) Mass on a spring



(b) Pendulum



(c) Falling object



(d) Pulsar

Figure 4.3: Original and counterfactual videos for experiments (a-d) obtained by decoding the original and intervened priors.

Intervention results. Now if we want to simulate a scenario where the mass is slowed down by air drag we can for example set $\gamma = 1/2\pi$ in eq. 4.14 to obtain the intervened prior $\theta_{int}(t) = e^{-t/2\pi}\cos(t)$ (fig. 4.2b, green). Decoding latent variables generated with this intervened prior results in a novel video where the mass slows down over time (fig. 4.3b, right). By changing the parameters in eq. 4.14 we could also obtain counterfactual videos of a mass oscillating with different strengths of air resistance (by intervening on γ), on different planets (by intervening on g), or with different amplitudes (by intervening on θ_0).

4.3.4 Falling Object

In this experiment an object is dropped from rest at an initial height and is left to accelerate and eventually hit the ground (fig. 4.3c, top). Assuming mass m, acceleration due to gravity g and assuming the object is released from rest, the equation of motion is determined by solving Newton's second law:

$$m\frac{\partial^2 y}{\partial t^2} = -mg \tag{4.15}$$

If the object is released from height y_0, eq. 4.15 is solved by

$$y = y_0 - \frac{1}{2}gt^2 \tag{4.16}$$

For simplicity we set $y_0 = 1$ and $g = 2$ to obtain the prior $y = 1 - t^2$.

Quantitative results. Fig 4.2c shows the quadratic prior (blue) matches the posterior (red). Table 4.2 (third row) shows that our method achieves the lowest MSE among all baselines, and a reconstruction accuracy comparable to the temporal VAE baseline.

Intervention results. Now if we want to simulate a scenario where the object is falling on the surface of Jupiter instead of on Earth's surface where $g_{\text{Jupiter}} = 2.53 g_{\text{Earth}}$, we can modify the prior as $y_{int} = 1 - 2.53t^2$ (fig. 4.2c, green) and decode it to obtain the corresponding video (fig. 4.3c, bottom) where the object now falls much faster. By intervening on eq. 4.16 we could also simulate many other

hypothetical scenarios including throwing the object up from the ground with a given initial speed and waiting for it to come back down, or seeing how adding the effects of air resistance would change the object's trajectory, or making the object bounce off when it hits the ground and after a few bounces slowly come to rest.

4.3.5 Pulsar

In this experiment we deal with image data from a pulsar, which is a rotating neutron star that emits periodic pulses of electromagnetic radiation. Here we use the image data of the Crab pulsar recorded at different phases in the UVB spectrum [107] (fig. 4.3d, top; based on fig. 2 of the cited paper). For each pulsar image we compute its intensity as the average of the pixel values across the image (fig. 4.4, top, black), and manually fit it with a Gaussian mixture model consisting of two Gaussians (fig. 4.4, top, red), corresponding to the two peaks in its emission profile. The prior intensity is thus

$$I(t) = A_1\mathcal{N}(t; \mu_1, \sigma_1^2) + A_2\mathcal{N}(t; \mu_2, \sigma_2^2) \tag{4.17}$$

where we set $\mu_1 = 0.055, \mu_2 = 0.5, \sigma_1 = 0.05, \sigma_2 = 0.08, A_1 = 0.125, A_2 = 0.085$.

Quantitative results. Fig 4.2d shows the prior (blue) matches the posterior (red). Table 4.2 (fourth row) shows that our method achieves the lowest MSE among all baselines and a reconstruction accuracy comparable to the temporal VAE baseline.

Intervention results. Now, if we want to visualise how the pulsar would look like at a different frequency, we can replace the intensity profile prior with the intensity profile for that frequency. For example, the intensity profile for the Crab pulsar at 1418 MHz is shown in fig. 4.4, bottom, in black [108] (fig. 9.2 of the cited paper). Again, we fit a Gaussian mixture model consisting of two Gaussians to this data (fig. 4.4, bottom, red) to obtain the intervention intensity prior as

$$I_{int}(t) = A_3\mathcal{N}(t; \mu_3, \sigma_3^2) + A_4\mathcal{N}(t; \mu_4, \sigma_4^2) \tag{4.18}$$

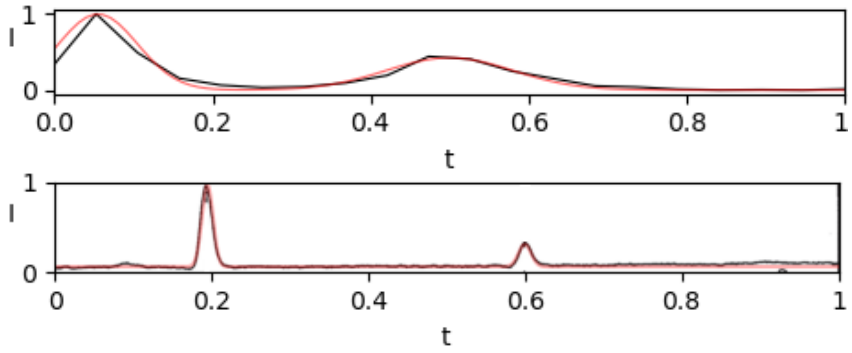


Figure 4.4: Top: Crab pulsar intensity data (black) and a double Gaussian mixture fit (red) at the original UVB frequency (top) and at the intervention frequency 1418 MHz (bottom).

where $\mu_3 = 0.195, \mu_4 = 0.6, \sigma_3 = 0.007, \sigma_4 = 0.008, A_3 = 0.0193, A_4 = 0.006$. The intervention curve is visualised in fig. 4.2d (green) and is decoded to obtain a corresponding series of intervention images for the new frequency (fig. 4.3d, bottom).

4.4 Discussion

While generally successful, the presented method possesses certain limitations:

1. Firstly, our method requires specifying the prior representing the physical equation for each video *a priori*, unlike humans who learn physics without this knowledge. This is because, if one were to learn both the physical equation prior as well as the physical latent variable from the video, the system would be underconstrained – it would learn any combination of the prior and the variable, which would not be guaranteed to be interpretable. To relax this constraint, one might employ interventions (just like humans do), such as launching a pendulum from different known heights and initial speeds, which might provide sufficient information to learn both the physical variable as well as the equation of motion.

2. Secondly, our method requires that the specified physical equation prior needs to exactly match the physics in each video. If the prior does not exactly match the object's equation of motion, the method may still converge, but

the latent space will no longer be physically interpretable and the generated videos will not correspond to expectations. For example, if one specifies the prior $y = 1 - t$ instead of $1 - t^2$ for a video of a falling object, the video frames will still get mapped to the $y = 1 - t$ prior but the variable y will no longer correspond to the object's position and a video generated by intervening on this prior will be incorrect. In the case of a video with periodic motion such as a pendulum, if one specifies an incorrect phase shift (e.g. a sine instead of a cosine), the method will try to match two different video frames to the same value of the latent variable, so the method might not be able to learn the variable correctly.

3. Thirdly, the method currently requires there to be a single object that moves in each video. If a video contains multiple objects, currently a single variable will be learned that entangles the motion of all objects. For the method to work for multiple objects, we speculate that one might have to add one variable per object with a separate equation of motion for each, and to intervene on the video to separate the motion of all the objects (for example, for a video of two masses each moving on a spring, first perturbing the first one and then the second one separately).

4.5 Conclusion

In this work we presented a novel method based on a VAE with an ANM prior to learn physical variables from videos, that can be used to imagine novel (counterfactual) physical scenarios. After learning variables related by a specified physical prior for each video, we were then able to intervene on the prior to synthesize videos of the same physics experiment under different conditions, such as a pendulum with added air drag or a mass on a spring with modified object mass and frequency. To achieve this, we proposed a novel method to approximate the non-linear ANM prior using a Gaussian Mixture Model where the shape of each Gaussian depends on the curvature of the prior. To efficiently optimise the VAE ELBO term we

derived a numerically stable and parallelizable Monte Carlo estimate of the KL divergence between the posterior and prior GMMs. We have shown that the method is successful in learning the correct variables and generating realistic counterfactual videos for different real physics videos.

While successful in application to videos that can be represented by a single variable, ultimately we would like to extend our method to more than one variable while still retaining identifiability guarantees. In the next chapter we will focus on a multivariate case where we will achieve identifiability via the use of equivariances.

5

Unsupervised Object Detection with Theoretical Guarantees

Contents

5.1 Introduction **51**
5.2 Method **53**
5.2.1 Autoencoder with CNN Encoder and Decoder 54
5.2.2 From Encoder Feature Maps to Latent Variables 55
5.2.3 From Latent Variables to Decoder Feature Maps 56
5.3 Theoretical Results **57**
5.4 Experimental Results **62**
5.4.1 Synthetic Experiments 64
5.4.2 CLEVR Experiments 65
5.5 Discussion **67**
5.6 Conclusion **69**

5.1 Introduction

Unsupervised object detection using deep neural networks is a long-standing area of research at the intersection of machine learning and computer vision. Its aim is to learn to detect objects from images without the use of training labels. Learning without supervision has multiple advantages, as obtaining labels for training data is often costly and time consuming, and in some cases may be impractical or unethical.

For example, in medical imaging, unsupervised object detection can help save specialists' time by automatically flagging suspicious abnormalities [109], and in autonomous driving it may help automatically detect pedestrians on a collision course with the vehicle [110]. It is thus important to understand and develop better unsupervised object detection methods.

While successful, current object detection methods are often empirical and possess few to no guarantees about their learned representations. In this chapter we aim to address this gap by designing the first unsupervised object detection method that we prove is guaranteed to learn the true object positions up to small shifts, and performing a detailed analysis of how the maximum errors of the learned object positions depend on the encoder and decoder receptive field sizes, the object sizes, and the sizes of the Gaussians used for rendering. This is especially important in sensitive domains such as medicine, where incorrectly detecting an object could be costly. Our method guarantees to detect any object that moves in a video or that appears at different locations in images, as long as the objects are distinct and the images are reconstructed correctly.

We base our unsupervised object detection method on an autoencoder with a convolutional neural network (CNN) encoder and decoder, and modify it to make it exactly translationally equivariant (sec. 5.2). This allows us to interpret the latent variables as object positions and lets us train the network without supervision. We then use the equivariance property to formulate and prove a theorem that relates the maximum position error of the learned latent variables to the size of the encoder and decoder receptive fields, the size of the objects, and the width of the Gaussian used in the decoder (sec. 5.3). Next, we derive corollaries describing the exact form of the maximum position error as a function of these four variables. These corollaries can be used as guidelines when designing unsupervised object detection networks, as they describe the guarantees of the learned object positions that can be obtained under different settings. We then perform synthetic and CLEVR-based [111] experiments to validate our theory (sec. 5.4). Finally, we discuss the implications of our results for designing reliable object detection methods (sec. 5.5).

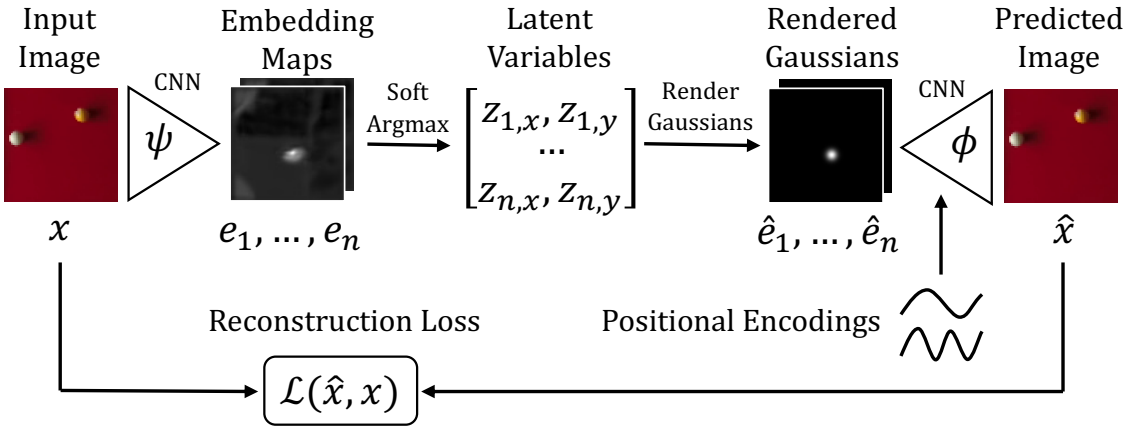


Figure 5.1: Network architecture. Encoder: (1) an image x is passed through a CNN ψ to obtain n embedding maps $e_1, ..., e_n$, (2) a maximum of each map is found using softargmax to obtain latent variables $[z_{1,x}, z_{1,y}, ..., z_{n,x}, z_{n,y}]$. Decoder: (1) Gaussians $\hat{e}_1, ..., \hat{e}_n$ are rendered at the positions given by the latent variables, (2) the Gaussian maps are concatenated with positional encodings and passed through a CNN ϕ to obtain the predicted image $\hat{x}$. Finally, x and $\hat{x}$ are used to compute reconstruction loss $\mathcal{L}(\hat{x}, x)$.

Concretely, the contributions of this chapter are:

1. The first unsupervised object detection method that is guaranteed to learn the true object positions up to small shifts.

2. A proof and detailed theoretical analysis of how the maximum position error of the method depends on the encoder and decoder receptive field sizes, object sizes, and widths of the Gaussians used in the rendering process.

3. Synthetic experiments, CLEVR-based experiments, and real video experiments validating our theoretical results up to precisions of individual pixels.

5.2 Method

In this section we describe the proposed method for unsupervised object detection with guarantees. On a high level, our architecture is based on an autoencoder that is fully equivariant to translations, which we achieve by making the encoder consist of a CNN followed by a soft argmax function to extract object positions, and making the decoder consist of a Gaussian rendering function followed by another

CNN to reconstruct an image from the object positions (fig. 5.1). In the following sections we describe the different parts of the architecture in detail.

5.2.1 Autoencoder with CNN Encoder and Decoder

We start with an autoencoder, a standard unsupervised representation learning model, consisting of an encoder network ψ that maps an image x to a low-dimensional latent variable z, followed by a decoder network ϕ that maps this variable back to an image $\hat{x}$, with the objective of minimising the difference between x and $\hat{x}$. Typically, the encoder and decoder networks are parametrised by multi-layer perceptrons (MLPs) or convolutional neural networks (CNNs) paired with fully-connected (FC) layers, however neither of these parametrisations by default can guarantee that the learned latent variables will correspond to the true object positions (because of the universal approximation ability of MLPs and FC layers hornik1989multilayer). To obtain such guarantees, we would thus like to modify the autoencoder to make it exactly translationally equivariant, that is, a shift of an object in the input image x should correspond to a proportional shift of the latent variable z, and a shift of the latent variable z should correspond to a shift in the predicted image $\hat{x}$.

We start with an autoencoder where the encoder and decoder are both CNNs. CNNs consist of layers computing the convolution between a feature map x and a filter F, defined in one dimension as

$$(x \star F)[i] = \sum_j x[j]F[j-i] \tag{5.1}$$

Intuitively, this corresponds to sliding the filter F across the feature map x and at each position of the filter i computing the dot product between the feature map x and the filter F. We can prove that convolutional layers are equivariant to translations, since

$$((\tau \circ x) \star F)[i] = \sum_j x[j-t]F[j-i] = \sum_j x[j]F[j-(i-t)] = \tau \circ (x \star F)[i] \tag{5.2}$$

where τ is the translation operator that translates a feature map by t pixels, and we have used the substitution $j \to j+t$ at the second equality. Therefore, the

encoder and decoder are both equivariant to translations, but this property only holds for translations of feature maps (i.e. spatial tensors).

5.2.2 From Encoder Feature Maps to Latent Variables

So far we have only worked with images and feature maps, but the latter do not directly express positions of any detected objects. It would be preferable to convert these feature maps into scalar variables that can be interpreted as object positions that are equivariant to image translations. To do this, we first define a translation τ of a (1D) feature map x and a translation τ' of a scalar z as

$$\tau(x)[i] = x[i-t], \quad \tau'(z) = z + t \tag{5.3}$$

where i is the position in the feature map x, τ shifts an image by t pixels, and τ' shifts a scalar by t units. To relate translations in feature maps to translations in latent variables, we can use a function that computes a scalar property of a feature map x, such as argmax, defined as $\text{argmax}(x) = \{i : x[j] \leq x[i] \ \forall j\}$. Using these definitions we can now prove the equivariance of argmax, i.e. that shifting the feature map x by τ corresponds to shifting the latent variable $\text{argmax}(x)$ by τ':

$$\begin{aligned}\text{argmax}(\tau \circ x) &= \{i : \tau \circ x[j] \leq \tau \circ x[i] \ \ \forall j\} = \{i : x[j-t] \leq x[i-t] \ \ \forall j\} \\ &= \{i + t : x[j] \leq x[i] \ \ \forall j\} = \text{argmax}(x) + t = \tau' \circ \text{argmax}(x)\end{aligned} \tag{5.4}$$

where at the first equality we use the definition of argmax, at the second equality we use the definition of τ (eq. 5.3, left), at the third equality we use the substitution $i \to i + t$, at the fourth equality we use the definition of argmax, and at the last equality we use the definition of τ' (eq. 5.3, right).

However, because the argmax operation is not differentiable, for neural network training we approximate it via a differentiable soft argmax function, defined in 2D as

$$\text{softargmax}(x) = \left(\frac{1}{I} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \left(i + \frac{1}{2}\right) \ \sigma_1\left(\frac{x}{\Theta}\right)[i,j], \ \frac{1}{J} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \left(j + \frac{1}{2}\right) \ \sigma_2\left(\frac{x}{\Theta}\right)[i,j] \right) \tag{5.5}$$

where σ is the softmax function defined in one dimension as $\sigma(x)[i] = \exp(x[i]) / \sum_j \exp(x[j])$, $\sigma_1(x)$ and $\sigma_2(x)$ is the softmax function evaluated

along the first and second dimensions of x, Θ is a temperature hyperparameter, $[i, j]$ is the image index, I is the image width, J is the image height, and the term $1/2$ ensures that the densities correspond to pixel centres. As the temperature Θ in eq. 5.5 approaches zero, softargmax reduces to the classical argmax function.

5.2.3 From Latent Variables to Decoder Feature Maps

Similar to mapping from encoder feature maps to latent variables, we would now like to relate shifts in latent variables z to shifts of decoder feature maps x. To do this, we can invert the action of the argmax operation. Because argmax is a many-to-one function, finding an exact inverse is not possible, but we can obtain a pseudo-inverse using the Dirac delta function defined as $\text{delta}(z)[i] = \delta(i - z)$. We can show that delta is a pseudo-inverse of argmax because $\text{argmax} \circ \text{delta} \circ z = i : \delta(j - z) \leq \delta(i - z)\ ; \forall j = z$. Now, similar to the argmax function, we can prove that the delta function is equivariant to the latent variable shift τ' on the input and the feature map shift τ on the output, i.e.

$$\text{delta}(\tau' \circ z)[i] = \delta(i - \tau' \circ z) = \delta(i - z - t) = \text{delta}(z)[i - t] = \tau \circ \text{delta}(z)[i] \tag{5.6}$$

where at the first equality we have used the definition of delta, at the second equality we have used the definition of τ' (eq. 5.3, right), at the third equality we have used the definition of delta, and at the last equality we have used the definition of τ (eq. 5.3, left).

Again, because the delta function is not differentiable, we can approximate it using a differentiable render function as

$$\text{render}(z)[i] = \mathcal{N}(i - z, \sigma^2) \tag{5.7}$$

where $\mathcal{N}(i - z, \sigma^2)$ is a Gaussian evaluated at $i - z$ with variance given by the hyperparameter σ^2. As the variance σ^2 in eq. 5.7 approaches zero, the render function reduces to the hard delta function.

Additionally, because the decoder is translationally equivariant, we also condition it on positional encodings of the size of the images to provide it with sufficient information to reconstruct different parts of the background, assuming the background

is static. Alternatively, if background is varying, the decoder can be conditioned on a randomly-sampled nearby video frame instead, which will provide information about the background but not the objects' positions (following Jakab et al. [112]).

We also note that since the latent variables z are ordered, this allows the encoder and decoder to learn to associate each variable with the semantics of each object and achieve successful reconstruction.

We thus now have all the elements we need to create an equivariant architecture where the encoder and decoder are defined, respectively, by

$$z = \text{softargmax} \circ \psi \circ x, \quad \hat{x}_t = \phi \circ \text{render} \circ z_t. \tag{5.8}$$

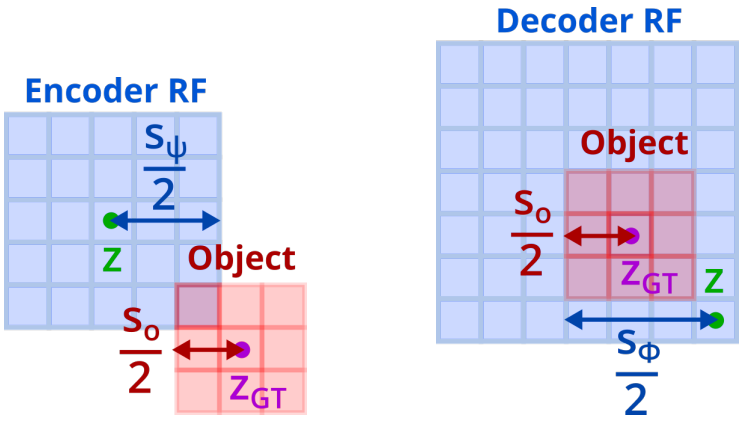
This is shown in fig. 5.1. Having designed an exactly translationally equivariant architecture now allows us to obtain theoretical guarantees about the learned latent variables, which we discuss next. [1]

5.3 Theoretical Results

In this section we present our main theorem stating the maximum bound on the position errors of the latent variables learned with our method in terms of the encoder and decoder receptive field sizes, the object size, and the Gaussian standard deviation (thm. 5.3.1). We continue by deriving specialised corollaries relating the maximum position error to the encoder receptive field size (cor. 5.3.2), decoder receptive field size (cor. 5.3.3), object size (cor. 5.3.4), and Gaussian standard deviation (cor. 5.3.5).

Theorem 5.3.1 (Error Bound)**.** *Consider a set of images $x \sim X$ with objects of size s_o, CNN encoder ψ with receptive field size s_ψ, CNN decoder ϕ with receptive field size s_ϕ, soft argmax function* softargmax*, rendering function* render *with Gaussian standard deviation σ_G and $\Delta_G \sim \mathcal{N}(0, \sigma_G^2)$, and latent variables z, composed as $z = \text{softargmax} \circ \psi \circ x$ and $\hat{x} = \phi \circ \text{render} \circ z$ (fig. 5.1). Assuming (1) the objects*

[1] We note that a similar architecture was proposed by Jakab et al. [112], with empirical success in keypoint detection. However, we derive our architecture by enforcing strict translation equivariance properties, which makes our theoretical results possible.



(a) Encoder Error **(b)** Decoder Error

Figure 5.2: Position errors. (a) Maximum position error due to encoder, given by $s_\psi/2 + s_o/2 - 1$. The maximum error occurs when the encoder and the object are as far away from each other as possible while still overlapping by one pixel. (b) Maximum position error due to decoder, given by $s_\phi/2 - s_o/2 + \Delta_G$. The maximum error occurs when some part of the Gaussian at position $z + \Delta_G$ is within the decoder receptive field (RF) but is as far away from the rendered object as possible.

are reconstructed at the same positions as in the original images, (2) each object appears in at least two different positions in the dataset, and (3) there are no two identical objects in any image, then the learned latent variables z correspond to the true object positions up to object permutations and maximum position errors Δ of

$$\Delta = \min\left(\frac{s_\psi}{2} + \frac{s_o}{2} - 1, \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G\right). \tag{5.9}$$

For proof see appendix B.1. Intuitively, the assumptions ensure that each latent variable corresponds to the position of each object in the image. The error in the learned object positions then arises from both the encoding and decoding process. In the encoding process, the maximum error occurs when the encoder and the object are as far away from each other as possible while still overlapping, i.e. $s_\psi/2+s_o/2-1$ (fig. 5.2a). Conversely, in the decoding process, the maximum error occurs when the rendered object and the latent variable are as far away from each other as possible while both still being inside the decoder receptive field, i.e. $s_\phi - s_o/2$ (fig. 5.2b). Additionally, there is an extra error of Δ_G as the latent variable is rendered by a Gaussian and the decoder can capture any part of this Gaussian. Finally, because we assume each object is reconstructed at the same position as

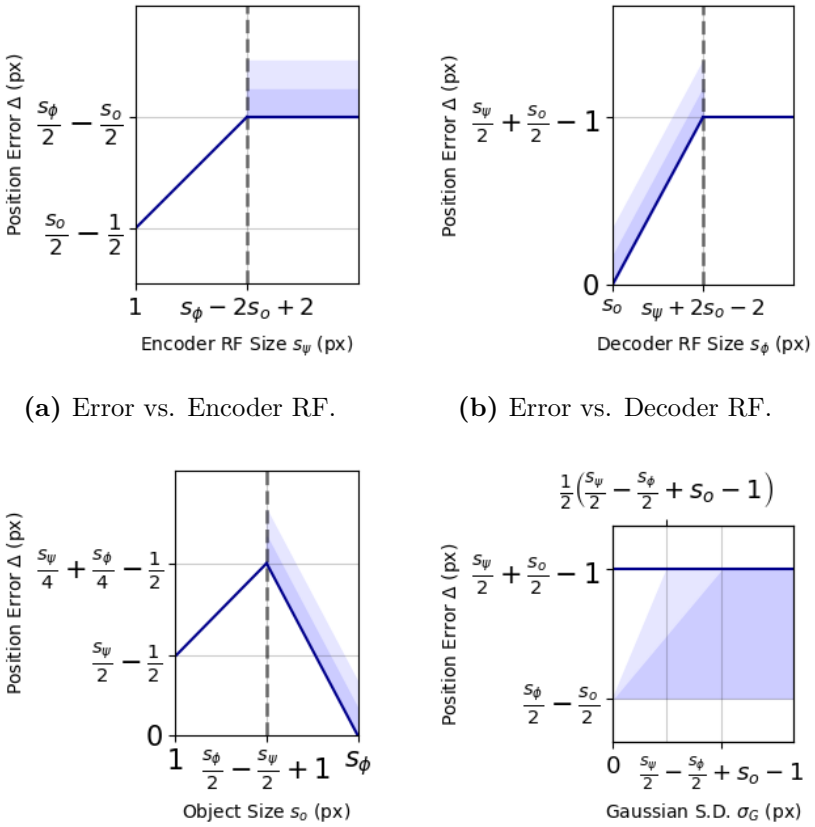
in the original image, the errors from the encoder and decoder must cancel each other out. Therefore, the overall maximum position error is given by the lower of the two expressions for the encoder and the decoder, leading to eq. 5.9. Next, we present corollaries relating this error bound to different factors.

Corollary 5.3.2 (Error Bound vs. Encoder RF Size)**.** *The maximum position error as a function of the encoder receptive field (RF) size s_ψ for a given s_ϕ, s_o, σ_G, is*

$$\Delta(s_\psi) = \begin{cases} \frac{s_\psi}{2} + \frac{s_o}{2} - 1 & \text{if} \quad 1 \leq s_\psi \leq s_\phi - 2s_o + 2, \\ \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G & \text{if} \quad s_\psi > s_\phi - 2s_o + 2. \end{cases}$$

For an illustration see fig. 5.3a. There are two regions of the curve (separated by a dashed line). In the left-most region, $s_\psi < s_\phi - 2s_o + 2$, the error is dominated by the encoder error, and in the right-most region, $s_\psi \geq s_\phi - 2s_o + 2$, the error is dominated by the decoder error. Initially, for $s_\psi = 1$, the error is given by $s_o/2 - 1/2$, because the 1×1 px encoder can match any pixel that is part of the object and so can be at most half of the object size away from the true object position that is at the centre of the object. As the encoder RF size increases up to $s_\phi - 2s_o + 2$, the position error increases linearly with it as $s_\psi/2 + s_o/2 - 1$, because now any part of the encoder RF can match any part of the object (fig. 5.2a). This bound is deterministic due to the deterministic encoding process.

At $s_\psi = s_\phi - 2s_o + 2$ (vertical dashed line in fig. 5.3a), the maximum errors from encoder and decoder both become equal to $s_\phi/2 - s_o/2$. For $s_\psi > s_\phi - 2s_o + 2$, the position error is dominated by the error from the decoder which is constant at $s_\phi/2 - s_o/2 + \Delta_G$ with $\Delta_G \sim \mathcal{N}(0, \sigma_G^2)$, and so even though the encoder RF size is increasing, this has no effect as the limiting factor is now the decoder. Due to the Gaussian rendering step in the decoding process, this bound is now probabilistic, and is distributed normally with variance σ_G^2.The results of corollary 5.3.2 can be extended to multiple objects with different sizes (see appendix B.2, cor. B.2.1)



(a) Error vs. Encoder RF. **(b)** Error vs. Decoder RF.

(c) Error vs. Object Size. **(d)** Error vs. Gaussian S.D.

Figure 5.3: Theoretical bounds for the maximum position error as a function of the encoder receptive field size s_ψ, decoder receptive field size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, as the remaining factors are fixed. Each bound consists of a region due to the encoder error (solid line) and the decoder error (probabilistic bound). Standard deviations are represented by shades of blue.

Corollary 5.3.3 (Error Bound vs. Decoder RF Size)**.** *The maximum position error as a function of the decoder receptive field (RF) size s_ϕ for a given s_ψ, s_o, σ_G, is*

$$\Delta(s_\phi) = \begin{cases} \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G & \text{if} \quad s_o \leq s_\phi < s_\psi + 2s_o - 2, \\ \frac{s_\psi}{2} + \frac{s_o}{2} - 1 & \text{if} \quad s_\phi \geq s_\psi + 2s_o - 2. \end{cases}$$

For an illustration see fig. 5.3b. Similar to corollary 5.3.2, there are two regions of the curve, one for $s_\phi < s_\psi + 2s_o - 2$ (left), where the error is dominated by the decoder error, and another for $s_\phi \geq s_\psi + 2s_o - 2$ (right), where the error is dominated by the encoder error. Note that this is opposite to cor. 5.3.2. Initially, for $s_\phi = s_o$, the decoder receptive field has the same size as the object, and so to achieve perfect

reconstruction it needs to be at the same position as the object, resulting in 0 position error plus any error Δ_G caused by the non-zero width of the Gaussian. As the decoder RF size increases up to $s_\psi + 2s_o - 2$, the position error increases linearly with it as $s_\phi/2 - s_o/2 + \Delta_G$, because now the object can be at an increasing number of positions within the decoder and still achieve perfect reconstructions (fig. 5.2b). At $s_\phi = s_\phi + 2s_o - 2$, the maximum errors from encoder and decoder both become equal to $s_\psi/2 + s_o/2 - 1$. For $s_\phi > s_\psi + 2s_o - 2$, the position error is dominated by the error from the encoder which is constant at $s_\psi/2 + s_o/2 - 1$, and so even though the decoder RF size is increasing, this has no effect as the limiting factor is now the encoder. Similar to corollary 5.3.2, the results of corollary 5.3.3 can be extended to objects with multiple different sizes (see appendix B.2, cor. B.2.2).

Corollary 5.3.4 (Error Bound vs. Object Size)**.** *The maximum position error as a function of the object size s_o for a given s_ψ, s_ϕ, σ_G, is*

$$\Delta(s_o) = \begin{cases} \frac{s_\psi}{2} + \frac{s_o}{2} - 1 & \text{if } \ 1 \leq s_o \leq \frac{s_\phi}{2} - \frac{s_\psi}{2} + 1, \\ \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G & \text{if } \ \frac{s_\phi}{2} - \frac{s_\psi}{2} + 1 < s_o \leq s_\phi. \end{cases}$$

For an illustration see fig. 5.3c. Again, there are two regions of the curve, one for $s_o < s_\phi/2 - s_\psi/2 + 1$ (left), where the error is dominated by the encoder error, and one for $s_o \geq s_\phi/2 - s_\psi/2 + 1$ (right), where the error is dominated by the decoder error. Initially, for $s_o = 1$, the error is given by $s_\psi/2 - 1/2$, because any pixel of the encoder receptive field can match the 1×1 px object and so the error can be at most half of the encoder receptive field size. As the object size increases up to $s_\phi/2 - s_\psi/2 + 1$, the position error increases linearly with it as $s_\psi/2 + s_o/2 - 1$, because now any part of the encoder RF can match any part of the object (fig. 5.2a). At $s_o = s_\phi/2 - s_\psi/2 + 1$, the maximum errors from encoder and decoder both become equal to $s_\psi/4 + s_\phi/4 - 1/2$. For $s_o > s_\phi/2 - s_\psi/2 + 1$, the position error is dominated by the error from the decoder and decreases linearly as $s_\phi/2 - s_o/2 + \Delta_G$, because now there is a decreasing number of positions where the object can still fit inside the decoder receptive field (fig. 5.2b). At $s_o = s_\phi$, the object reaches the same size as the decoder, and thus the position error decreases to 0 with an

additional error due to the width of the Gaussian, Δ_G. Interestingly, the triangular shape of the error curve means that small and large objects will both incur small position errors, while medium sized objects will incur higher errors.

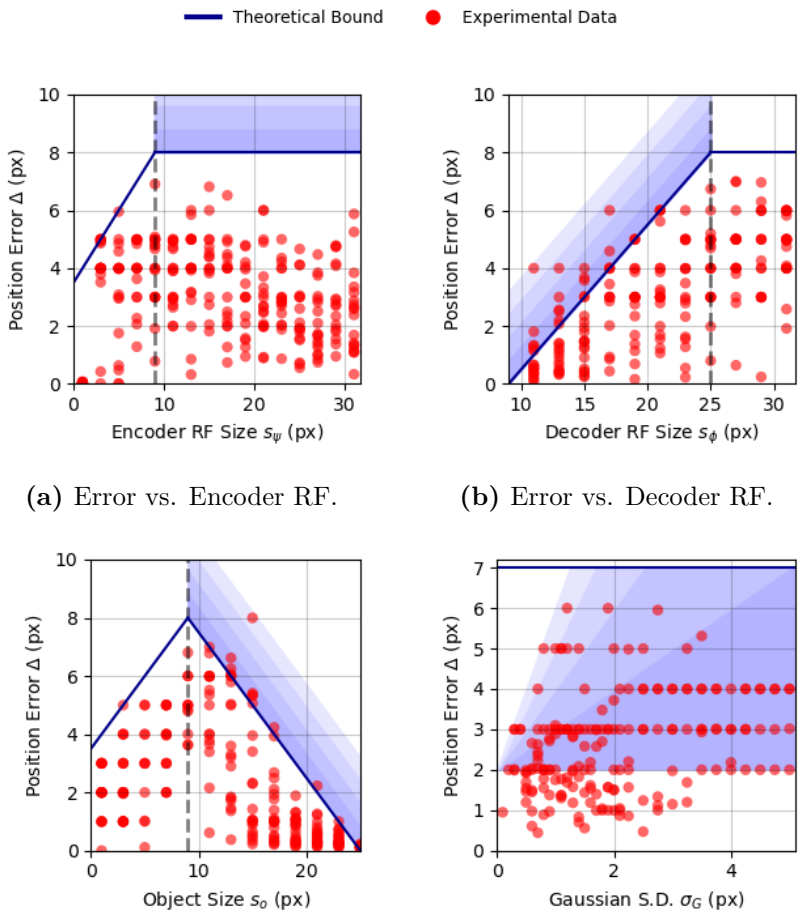
Corollary 5.3.5 (Error Bound vs. Gaussian Size)**.** *The maximum position error as a function of the Gaussian standard deviation σ_G for a given s_ψ, s_ϕ, s_o, is*

$$\Delta(\sigma_G) = \begin{cases} \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G & \text{if } \sigma_G < \frac{s_\psi}{2} - \frac{s_\phi}{2} + s_o - 1, \\ \frac{s_\psi}{2} + \frac{s_o}{2} - 1 & \text{if } \sigma_G \geq \frac{s_\psi}{2} - \frac{s_\phi}{2} + s_o - 1. \end{cases}$$

For an illustration see fig. 5.3d. Firstly, there is an overall maximum bound for the position error due to the encoder, given by the constant $s_\psi/2 + s_o/2 - 1$, which is independent of the Gaussian standard deviation. Then, initially for $\sigma_G = 0$, the rendered Gaussian is effectively a delta function, and so the position error is dominated by the decoder error given by $s_\phi/2 - s_o/2$, which describes the maximum distance between the object and the delta function with both of them fitting inside the decoder receptive field (fig. 5.3b). As the Gaussian standard deviation increases, the position error increases linearly as $s_\phi/2 - s_o/2 + \Delta_G$ with $\Delta_G \sim \mathcal{N}(0, \sigma_G^2)$. Then, depending on what part of the Gaussian the decoder is convolved with, there are different bounds for the maximum position error. If the decoder is convolved with a part of the Gaussian that is within n standard deviations of its centre, the maximum position error increases linearly as $s_\phi/2 - s_o/2 + n\sigma_G$ up to $\sigma_G = (s_\psi/2 - s_\phi/2 + s_o - 1)/n$, after which point the position error becomes dominated by the encoder value of $s_\psi/2 + s_o/2 - 1$. In fig. 5.3d, the maximum position error bound when the decoder is convolved with a part of the Gaussian within its first and second standard deviations, is denoted by darker and lighter shades of blue, respectively.

5.4 Experimental Results

In this section we validate our theoretical results on synthetic experiments (sec. 5.4.1) and CLEVR data (sec. 5.4.2). Additional real video experiments are in app.



(a) Error vs. Encoder RF. **(b)** Error vs. Decoder RF.

(c) Error vs. Object Size. **(d)** Error vs. Gaussian S.D.

Figure 5.4: Synthetic experiment results showing position error as a function of the encoder receptive field size s_ψ, decoder receptive field size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, as the remaining factors are fixed to $s_\psi = 9, s_\phi = 25, s_o = 9, \sigma_G = 0.8$ (in a–c) or to $s_\psi = 9, s_\phi = 11, s_o = 7$ (in d). Theoretical bounds are denoted by a blue line (with 4 shaded regions denoting 1 to 4 standard deviations of the probabilistic bound) and experimental results by red dots.

B.5. We first validate corollaries 5.3.2-5.3.4 via synthetic experiments in sec. 5.4.1, demonstrating very high agreement up to sizes of individual pixels. We then apply our method to CLEVR-based [111] data containing multiple objects of different sizes in varying scenes (sec. 5.4.2) and show that compared to current SOTA object detection methods (SAM [11], CutLER [12]), only our method predicts positions within theoretical bounds.

5.4.1 Synthetic Experiments

In this section we validate corollaries 5.3.2-5.3.5 via synthetic experiments. Our dataset consists of a small white square on a black background. In each experiment we fix all but one of the encoder RF size s_ψ, decoder RF size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, and vary the remaining variable. We perform each experiment 20 times, corresponding to 20 random initializations of the trained parameters, and record the position error Δ as the difference between the predicted object position z and the ground truth object position z_{GT}. For more details see appendix B.3.1.

Position Error vs. Encoder RF Size. In this experiment we aim to empirically validate corollary 5.3.2, by measuring the experimental position errors as a function of the encoder receptive field size. We vary the encoder RF sizes $s_\psi \in \{1, 3, \dots, 31\}$ and fix $s_\phi = 25, s_o = 9, \sigma_G = .8$ and record position errors Δ. We visualise the data points (red) and the theoretical bounds (blue) in fig. 5.4a. We can observe that all the data points lie at or below the theoretical boundary, which validates corollary 5.3.2. In particular, we observe that the deterministic boundary in the region to the left of the dashed line (corresponding to the encoder bound) is well respected, with some of the trained networks achieving exactly the maximum error predicted by theory.

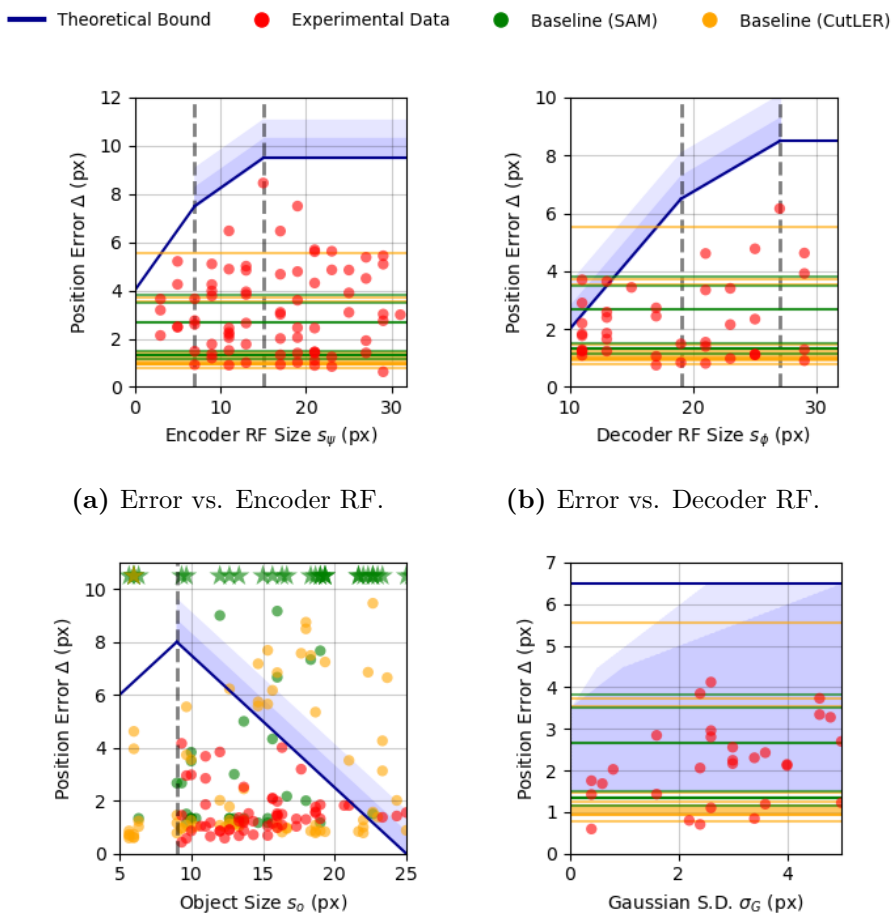
Position Error vs. Decoder RF Size. In this experiment we aim to validate corollary 5.3.3 by measuring the experimental position errors as a function of the decoder receptive field size. We vary the decoder RF sizes $s_\phi \in \{1, 3, \dots, 31\}$ and fix $s_\phi = 9, s_o = 9, \sigma_G = .8$ and record position errors Δ. We visualise the results in fig. 5.4b. The figure shows the theory to be a strong fit to the data, validating corollary 5.3.3. In particular, we note that the data points fit the Gaussian distribution in the decoder part of the curve (left of the dashed line) and are very close to (1 px below) the deterministic upper bound in the encoder part of the curve (right).

Position Error vs. Object Size. In this experiment we aim to validate corollary 5.3.4 by measuring the experimental position errors as a function of the object size. We vary the object sizes $s_o \in \{1, 3, \ldots, 25\}$ and fix $s_\psi = 9, s_\phi = 25, \sigma_G = .8$ and record position errors Δ. We visualise the results in fig. 5.4c. As all the data points lie at or below the theoretical boundary, this validates corollary 5.3.4. We note that the empirical distribution of errors follows very closely the shape of the theoretical bound, very strictly on the left side of the dashed line (encoder bound) and according to the distribution predicted on the right side (decoder bound).

Position Error vs. Gaussian Size. In this experiment we aim to validate corollary 5.3.5 by measuring the experimental position errors as a function of the Gaussian standard deviation. We vary the Gaussian standard deviations $\sigma_G \in \{0.1, 0.2, \ldots, 2.1, 2.25, 2.5, ..., 5\}$, fixing $s_\psi = 9, s_\phi = 11, s_o = 7$ and record position errors Δ. We visualise the data points (red) and the theoretical bounds (blue) in fig. 5.4d. As all the data points lie at or below the theoretical boundary, this validates corollary 5.3.5. In particular, we note that all the data points lie below the encoder bound (solid blue line), and all the data points lie within the bound denoted by four standard deviations away from the Gaussian. This means that in practice, the decoder can be convolved with any part of the Gaussian that lies within 4 standard deviations (corresponding to 3.2 px) from its centre. We also note that as the Gaussian standard deviation increases, the position error increases as expected, denoted by the positive slope of the data points between the third and fourth standard deviations (lightest shade of blue).

5.4.2 CLEVR Experiments

In this section we validate our theory on CLEVR-based [111] image data of 3D scenes. Our dataset consists of 3 spheres of different colours at random positions, with a range of sizes due to perspective distortion. We train and evaluate each experiment similarly to those in sec. 5.4.1, recording position errors for the learned objects, and compare our results with SAM [11] and CutLER [12] baselines. We



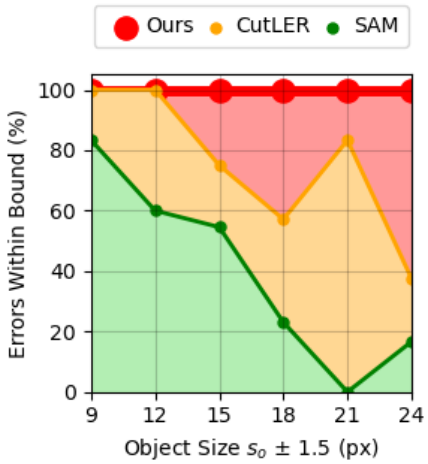
(a) Error vs. Encoder RF. **(b)** Error vs. Decoder RF.

(c) Error vs. Object Size. **(d)** Error vs. Gaussian S.D.

Figure 5.5: CLEVR experiment results showing position error as a function of the encoder receptive field size s_ψ, decoder receptive field size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, as the remaining factors are fixed to $s_\psi = 9, s_\phi = 25, s_o \in [6, 10], \sigma_G = 0.8$ for (a)–(c) and to $s_\psi = 5, s_\phi = 13, s_o \in [6, 10]$ for (d). Theoretical bounds are denoted in blue, experimental results in red, SAM baseline in green, and CutLER baseline in orange.

compute the theoretical bounds according to our theory in sec. 5.3 and app. B.2, and visualise the results in figs. 5.5a–5.5c. For details see app. B.3.2. For experiments with different shapes see app. B.4.

Once again, the experimental results demonstrate high agreement with our theory, now even for more complex images with multiple objects and a range of object sizes (fig. 5.5, red, blue). Furthermore, while the SAM and CutLER baselines generally achieve low position errors, this is not guaranteed, and in some cases their errors are much higher than our bound (fig. 5.5, green, orange). We report the proportion of position errors from fig. 5.5c that lie within 2 standard deviations



(a) Results visualisation.

Errors Within Bound (%)

	Object Size ± 1.5 (px)						
Method	All	9	12	15	18	21	24
Ours	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
CutLER	78.4	**100.0**	**100.0**	75.0	57.1	83.3	37.5
SAM	43.5	83.3	60.0	54.5	23.1	0.0	16.7

(b) Results table.

Figure 5.6: Proportion of position errors within 2 standard deviations of the theoretical bound (%), reported for different object sizes and methods. Results from table (b) are visualised in plot (a).

of our theoretical bound in table 5.6b and fig. 5.6a, showing that compared to SOTA object detection methods, only for our method are the position errors always guaranteed to be within our theoretical bound.

5.5 Discussion

In light of our theoretical results, in this section we present some conclusions that can be drawn when designing new unsupervised object detection methods:

1. If the size of the objects that will be detected is known, to minimise the error on the learned object positions, one should aim to design the decoder receptive field size to be as small as possible while still encompassing the object. As the

decoder RF grows beyond the object size, the error bound increases linearly with it up to a certain point (fig. 5.3b).

2. To minimise the error stemming from the encoder for a given object size, the encoder RF size should be kept as small as possible while still detecting the object (the RF size may be smaller than the object size), as again the error bound grows linearly with it up to a certain point (fig. 5.3a).

3. To minimise the error, the width of the rendering Gaussian should be kept as small as possible while still permitting gradient flow, as increasing it even slightly may result in a dramatic increase to the decoder term of the position error (fig. 5.3d). This is because, in practice, the decoder is able to detect parts of the Gaussian that are even 4 standard deviations away from its centre (fig. 5.4d).

4. In the case that one does not know *a priori* the exact size of the objects to be detected, one can still design a network that minimises the position errors for a given range of sizes. In that case, one should set up the decoder RF size to be as close as possible to the size of the largest object, and keep the encoder RF size as small as possible while still detecting all objects. The position errors for different object sizes will then be distributed according to the curve in fig. 5.3c, where the smallest and largest objects will achieve lowest errors and medium-size objects will achieve the greatest error, approximately given by a half of the average of the encoder and decoder RF sizes.

Finally, we discuss some limitations of our method. Firstly, the method can only detect dynamic objects, for example if they move in a video or if they appear at multiple locations in images. Secondly, in its current form the method learns representations that can not be used for videos with different backgrounds than the one used at training time; however, this can be overcome by conditioning the decoder on an unrelated video frame instead of the positional encodings, as in Jakab et al. [112]. Thirdly, the guarantees of our method are conditional on the images being successfully reconstructed, which depends on the network architecture and optimisation method.

5.6 Conclusion

We have presented the first unsupervised object detection method that is provably guaranteed to recover the true object positions up to small shifts. We proved that the object positions are learned up to a maximum error related to the encoder and decoder receptive field sizes, the object sizes, and bandwidth of the Gaussians used to render the objects. We derived expressions for how the position error depends on each of these factors and performed synthetic experiments that validated our theory up to sizes of individual pixels. We then performed experiments on CLEVR-based data, showing that unlike current SOTA methods, the position errors our method are always guaranteed to be within our theoretical bounds. We hope our work will provide a starting point for more research into object detection methods that possess theoretical guarantees, which are lacking in current practice.

In the next chapter we will summarise the contributions of the previous chapters and outline some directions for future work.

6 Conclusion

In this thesis we have investigated how to make current deep learning models more interpretable, enjoy better identifiability guarantees, and be more generalisable to novel out-of-distribution settings, using the framework of causality. To do this, the core idea is to incorporate various assumptions into unsupervised deep learning architectures, such as restrictions on the mechanism function class, the use of controlled settings with few variables, and enforcement of architectural assumptions such as equivariances. Doing this allowed us to obtain methods that learn representations that are meaningful, possess guarantees, and can be extrapolated to unseen settings, properties which are often difficult to obtain with current deep learning systems.

6.1 Summary

We started in chapter 3 by first developing a general unsupervised learning model based on a VAE with an Additive Noise Model prior for learning to represent images and videos using a causal graph with sparse edges and nonlinear mechanisms. We have successfully applied it to video game based data to learn interpretable object positions and mechanisms describing the object's motion, and by intervening on the graph we were able to extrapolate the video to the future.

Seeking better learning guarantees, in chapter 4 we reduced the problem to a single variable that is a function on time, and used a VAE with a non-linear ANM prior to learn physically interpretable variables describing from videos of physics experiments. We showed the model's ability to generalise by intervening on the variable and generating physically plausible videos of the experiments under different environmental conditions never observed during training (such as adding air drag, changing object mass, or performing the experiment on a different planet).

Relaxing the single-variable assumption to a two-variable case while still retaining learning guarantees, in chapter 5 we developed the first unsupervised object detection method that is theoretically guaranteed to recover the true object positions up to quantifiable small shifts. We proved that the learned variables correspond to the true object positions up to small shifts related to the encoder and decoder receptive field sizes, the object sizes, and the widths of the Gaussians used in the rendering process, and performed experiments validating our theoretical predictions up to a precision of individual pixels.

6.2 Limitations and Challenges

Although overall successful, the methods presented in the preceding chapters possess several limitations and challenges in terms of the settings they can be applied to as well as their identifiability guarantees, which we discuss here.

We start with the causal graph learning method from chapter 3. This method, while in theory able to learn arbitrary nonlinear causal graphs, in practice requires strong priors on the causal variables and mechanisms in order to obtain identifiability. For example, for the method to learn object positions of a video game character moving at a constant speed, the method requires constraining the causal mechanisms to be linear. Other future challenges include dealing with multiple objects, non-static backgrounds, and more complex causal graphs.

We proceed with the method for learning physically interpretable variables in chapter 4. While being good at learning interpretable physical variables and generating high fidelity realistic counterfactual videos in physics experiment settings,

to achieve identifiability, the method is currently limited to a single variable for each video that is a function of time, and requires specifying a physical prior for that video. It also requires that each video contains a single object as well as a static background. One challenge is therefore extending the method to multiple objects, in which case we speculate that the two objects will need to be intervened on separately in order to achieve identifiability. Another challenge is extending the method to non-static backgrounds, and possibly learning the physical priors directly from data.

Finally, we discuss the method for unsupervised object detection with theoretical guarantees presented in chapter 5. While able to obtain guarantees about the learned variables representing object positions, the method requires the videos to have a static background and the objects to move independently in order to achieve identifiability. One challenge would thus be to extend the method to varying backgrounds and more complex scenarios.

6.3 Future Work

As causal representation learning is still a very new field, many open problems remain about the best way to apply techniques from causality to machine learning, and vice versa. Here we outline a few research directions.

Firstly, most current visual causal representation learning methods deal with synthetic data such as images with simple shapes [50, 52]. While some progress has been made to make these more realistic [113], there is as of yet no method that would perform causal learning from complex real world videos. While it is difficult to obtain identifiability guarantees in this unconstrained setting, researchers should investigate methods that deal with more complex image and video data, albeit with possibly weaker identifiability guarantees.

Next, there is potential to apply techniques from causality to the field of reinforcement learning to improve the reasoning capabilities of agents [16]. An agent with a causal model of the world would be able to reason about hypothetical scenarios and thus be able to take better actions to achieve its goals. The recent field of causal reinforcement learning aims to deal with these questions.

Yet another direction is scaling current causal learning methods to more variables and more complex causal mechanisms [16]. Many of the current methods only deal with a few latent variables and often assume linear causal mechanisms [114]. Scaling these to complex graphs of tens of variables and the ability to learn non-linear causal mechanisms while still maintaining identifiability would make these methods better applicable to many real-world settings.

In conclusion, as the limitations of current AI systems are becoming more apparent, causality offers a compelling framework to help address some of their issues. In a world where AI is being adopted in more and more parts of society, causality could help make it a safer, more trustworthy, and more interpretable companion.

References

[1] OpenAI. "GPT-4 Technical Report". In: *ArXiv* abs/2303.08774 (2023). URL: https://api.semanticscholar.org/CorpusID:257532815.

[2] James Betker et al. *Improving Image Generation with Better Captions*. 2023. URL: https://cdn.openai.com/papers/dall-e-3.pdf.

[3] John Jumper, Richard Evans, Alexander Pritzel, et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (2021), pp. 583–589. URL: https://doi.org/10.1038/s41586-021-03819-2.

[4] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, et al. "Mastering Atari, Go, chess and shogi by planning with a learned model". In: *Nature* 588 (2020), pp. 604–609. URL: https://doi.org/10.1038/s41586-020-03051-4.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[6] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". eng. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[7] Gordon E. Moore. "Cramming more components onto integrated circuits". In: *Electronics* 38.8 (1965).

[8] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[9] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[10] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[11] Alexander Kirillov et al. "Segment anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4015–4026.

[12] Xudong Wang et al. "Cut and learn for unsupervised object detection and instance segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 3124–3134.

[13] Judea Pearl. *Causality*. 2nd ed. Cambridge: Cambridge University Press, 2009. URL: https://www.cambridge.org/core/books/causality/B0046844FAE10CBF274D4ACBDAEB5F5B (visited on 06/12/2022).

[14] Robert J. LaLonde. "Evaluating the Econometric Evaluations of Training Programs with Experimental Data". In: *The American Economic Review* 76.4 (1986). Publisher: American Economic Association, pp. 604–620. URL: https://www.jstor.org/stable/1806062 (visited on 06/12/2022).

[15] Karen Sachs et al. “Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data”. In: *Science* 308.5721 (Apr. 2005). Publisher: American Association for the Advancement of Science, pp. 523–529. URL: https://www.science.org/doi/full/10.1126/science.1105809 (visited on 06/12/2022).

[16] Bernhard Schölkopf et al. “Toward Causal Representation Learning”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[17] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[18] Dominik Rothenhäusler et al. “Anchor regression: Heterogeneous data meet causality”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 83.2 (2021), pp. 215–246.

[19] Yoshua Bengio et al. “A meta-transfer objective for learning to disentagle causal mechanisms”. In: *8th International Conference on Learning Representations (ICLR 2020)*. International Conference on Learning Representations (ICLR), 2020. URL: https://publications.polymtl.ca/53511/.

[20] Sara Magliacane et al. “Domain adaptation by using causal inference to predict invariant conditional distributions”. In: *Advances in neural information processing systems* 31 (2018).

[21] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. “Learning the difference that makes a difference with counterfactually-augmented data”. In: *arXiv preprint arXiv:1909.12434* (2019).

[22] Judea Pearl. “Causal Diagrams for Empirical Research”. In: *Biometrika* 82.4 (1995), pp. 669–688. URL: http://www.jstor.org/stable/2337329 (visited on 12/01/2023).

[23] Donald B Rubin. “Estimating causal effects of treatments in randomized and nonrandomized studies”. In: *Journal of Educational Psychology* 66.5 (1974), pp. 688–701. URL: https://doi.org/10.1037/h0037350.

[24] Liuyi Yao et al. “A Survey on Causal Inference”. In: *ACM Trans. Knowl. Discov. Data* 15.5 (May 2021). URL: https://doi.org/10.1145/3444944.

[25] Clark Glymour, Kun Zhang, and Peter Spirtes. “Review of Causal Discovery Methods Based on Graphical Models”. In: *Frontiers in Genetics* 10 (2019). URL: https://www.frontiersin.org/articles/10.3389/fgene.2019.00524.

[26] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[27] David Maxwell Chickering. “Optimal structure identification with greedy search”. In: *The Journal of Machine Learning Research* 3.null (Mar. 2003), pp. 507–554. URL: https://doi.org/10.1162/153244303321897717 (visited on 06/12/2022).

[28] Shohei Shimizu et al. “A Linear Non-Gaussian Acyclic Model for Causal Discovery”. In: *Journal of Machine Learning Research* 7.72 (2006), pp. 2003–2030. URL: http://jmlr.org/papers/v7/shimizu06a.html.

[29] Patrik Hoyer et al. "Nonlinear causal discovery with additive noise models". In: *Advances in Neural Information Processing Systems*. Vol. 21. Curran Associates, Inc., 2008. URL: https://papers.nips.cc/paper/2008/hash/f7664060cc52bc6f3d620bcedc94a4b6-Abstract.html (visited on 06/12/2022).

[30] Kun Zhang and Aapo Hyvarinen. "On the identifiability of the post-nonlinear causal model". In: *arXiv preprint arXiv:1205.2599* (2012).

[31] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[32] Danilo Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1530–1538. URL: https://proceedings.mlr.press/v37/rezende15.html.

[33] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536. URL: https://doi.org/10.1038/323533a0.

[34] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[35] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423.

[36] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. Tech. rep. arXiv:1312.6114. arXiv:1312.6114 [cs, stat] type: article. arXiv, May 2014. URL: http://arxiv.org/abs/1312.6114 (visited on 06/12/2022).

[37] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Sy2fzU9gl.

[38] Hyunjik Kim and Andriy Mnih. "Disentangling by Factorising". en. In: *Proceedings of the 35th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2018, pp. 2649–2658. URL: https://proceedings.mlr.press/v80/kim18b.html (visited on 06/09/2022).

[39] Casper Kaae Sønderby et al. "Ladder Variational Autoencoders". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016. URL: https://papers.nips.cc/paper/2016/hash/6ae07dcb33ec3b7c814df797cbda0f87-Abstract.html (visited on 06/08/2022).

[40] Arash Vahdat and Jan Kautz. "NVAE: A Deep Hierarchical Variational Autoencoder". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 19667–19679. URL: https://proceedings.neurips.cc/paper/2020/hash/e3b21256183cf7c2c7a66be163579d37-Abstract.html (visited on 06/09/2022).

[41] Rewon Child. "Very Deep {VAE}s Generalize Autoregressive Models and Can Outperform Them on Images". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=RLRXCV6DbEJ.

[42] Pierre Comon. "Independent component analysis, A new concept?" In: *Signal Processing* 36.3 (1994). Higher Order Statistics, pp. 287–314. URL: https://www.sciencedirect.com/science/article/pii/0165168494900299.

[43] Francesco Locatello et al. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 4114–4124. URL: https://proceedings.mlr.press/v97/locatello19a.html.

[44] Quanhan Xi and Benjamin Bloem-Reddy. "Indeterminacy in Generative Models: Characterization and Strong Identifiability". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Ed. by Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent. Vol. 206. Proceedings of Machine Learning Research. PMLR, 25–27 Apr 2023, pp. 6912–6939. URL: https://proceedings.mlr.press/v206/xi23a.html.

[45] Aapo Hyvarinen and Hiroshi Morioka. "Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/d305281faf947ca7acade9ad5c8c818c-Paper.pdf.

[46] Ilyes Khemakhem et al. "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 2207–2217. URL: https://proceedings.mlr.press/v108/khemakhem20a.html.

[47] David A. Klindt et al. "Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=EbIDjBynYJ8.

[48] Kartik Ahuja, Jason Hartford, and Yoshua Bengio. "Properties from mechanisms: an equivariance perspective on identifiable representation learning". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=g5ynW-jMq4M.

[49] Francesco Locatello et al. "Weakly-Supervised Disentanglement Without Compromises". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 6348–6359. URL: https://proceedings.mlr.press/v119/locatello20a.html.

[50] Julius Von Kügelgen et al. "Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. 2021. URL: https://openreview.net/forum?id=4pf_pOo0Dt.

[51] Johann Brehmer et al. "Weakly supervised causal representation learning". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=dz79MhQXWvg.

[52] Phillip Lippe et al. "CITRIS: Causal Identifiability from Temporal Intervened Sequences". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 13557–13603. URL: https://proceedings.mlr.press/v162/lippe22a.html.

[53] Kartik Ahuja et al. "Interventional Causal Representation Learning". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 372–407. URL: https://proceedings.mlr.press/v202/ahuja23a.html.

[54] Sebastien Lachapelle et al. "Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA". In: *First Conference on Causal Learning and Reasoning*. 2022. URL: https://openreview.net/forum?id=dHsFFekd_-o.

[55] Sebastien Lachapelle et al. "Synergies between Disentanglement and Sparsity: Generalization and Identifiability in Multi-Task Learning". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 18171–18206. URL: https://proceedings.mlr.press/v202/lachapelle23a.html.

[56] Gemma Elyse Moran et al. "Identifiable Deep Generative Models via Sparse Decoding". In: *Transactions on Machine Learning Research* (2022). URL: https://openreview.net/forum?id=vd0onGWZbE.

[57] Taco Cohen and Max Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2990–2999. URL: https://proceedings.mlr.press/v48/cohenc16.html.

[58] Robert Gens and Pedro M Domingos. "Deep Symmetry Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf.

[59] Taco S. Cohen and Max Welling. "Steerable CNNs". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=rJQKYt5ll.

[60] Taco Cohen et al. “Gauge Equivariant Convolutional Networks and the Icosahedral CNN”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 1321–1330. URL: https://proceedings.mlr.press/v97/cohen19d.html.

[61] Maurice Weiler and Gabriele Cesa. “General E(2)-Equivariant Steerable CNNs”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf.

[62] Joao F. Henriques and Andrea Vedaldi. *Warped Convolutions: Efficient Invariance to Spatial Transformations*. 2017. URL: https://openreview.net/forum?id=BkmM8Dceg.

[63] Diego Marcos, Michele Volpi, and Devis Tuia. “Learning rotation invariant convolutional filters for texture classification”. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, Dec. 2016. URL: https://doi.org/10.1109%5C%2Ficpr.2016.7899932.

[64] Diego Marcos et al. “Rotation Equivariant Vector Field Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. URL: https://doi.org/10.1109%5C%2Ficcv.2017.540.

[65] Junying Li et al. “Deep Rotation Equivariant Network”. In: *Neurocomputing* 290 (2018), pp. 26–33. URL: https://www.sciencedirect.com/science/article/pii/S0925231218301644.

[66] Sander Dieleman, Kyle W. Willett, and Joni Dambre. “Rotation-invariant convolutional neural networks for galaxy morphology prediction”. In: *Monthly Notices of the Royal Astronomical Society* 450.2 (Apr. 2015), pp. 1441–1459. eprint: https://academic.oup.com/mnras/article-pdf/450/2/1441/3022697/stv632.pdf. URL: https://doi.org/10.1093/mnras/stv632.

[67] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. “Exploiting Cyclic Symmetry in Convolutional Neural Networks”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1889–1898. URL: https://proceedings.mlr.press/v48/dieleman16.html.

[68] Jiaming Han et al. *ReDet: A Rotation-equivariant Detector for Aerial Object Detection*. 2021. arXiv: 2103.07733 [cs.CV].

[69] Nicolas Pielawski et al. *CoMIR: Contrastive Multimodal Image Representation for Registration*. 2020. arXiv: 2006.06325 [cs.CV].

[70] Deepak K. Gupta, Devanshu Arya, and Efstratios Gavves. *Rotation Equivariant Siamese Networks for Tracking*. 2020. arXiv: 2012.13078 [cs.CV].

[71] Daniel E. Worrall et al. *Harmonic Networks: Deep Translation and Rotation Equivariance*. 2017. arXiv: 1612.04642 [cs.CV].

[72] Angjoo Kanazawa, Abhishek Sharma, and David W. Jacobs. "Locally Scale-Invariant Convolutional Neural Networks". In: *CoRR* abs/1412.5104 (2014). arXiv: 1412.5104. URL: http://arxiv.org/abs/1412.5104.

[73] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. *Scale-Equivariant Steerable Networks*. 2020. arXiv: 1910.11093 [cs.CV].

[74] Jongmin Lee, Byungjin Kim, and Minsu Cho. *Self-Supervised Equivariant Learning for Oriented Keypoint Detection*. 2022. arXiv: 2204.08613 [cs.CV].

[75] Mohamed Adel Musallam et al. *Leveraging Equivariant Features for Absolute Pose Regression*. 2022. arXiv: 2204.02163 [cs.CV].

[76] Jiafei Duan et al. "A Survey on Machine Learning Approaches for Modelling Intuitive Physics". In: *arXiv preprint arXiv:2202.06481* (2022).

[77] Adam Lerer, Sam Gross, and Rob Fergus. "Learning Physical Intuition of Block Towers by Example". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 430–438. URL: https://proceedings.mlr.press/v48/lerer16.html.

[78] Jiajun Wu et al. "Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/d09bf41544a3365a46c9077ebb5e35c3-Paper.pdf.

[79] Jiajun Wu et al. "Physics 101: Learning physical object properties from unlabeled videos". In: *British Machine Vision Conference*. 2016.

[80] Kexin Yi* et al. "CLEVRER: Collision Events for Video Representation and Reasoning". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HkxYzANYDB.

[81] Nicholas Watters et al. "Visual Interaction Networks: Learning a Physics Simulator from Video". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8cbd005a556ccd4211ce43f309bc0eac-Paper.pdf.

[82] Peter Battaglia et al. "Interaction Networks for Learning about Objects, Relations and Physics". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/3147da8ab4a0437c15ef51a5cc7f2dc4-Paper.pdf.

[83] Katerina Fragkiadaki et al. *Learning Visual Predictive Models of Physics for Playing Billiards*. 2015. URL: https://arxiv.org/abs/1511.07404.

[84] Roozbeh Mottaghi et al. "Newtonian Scene Understanding: Unfolding the Dynamics of Objects in Static Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[85] Jiajun Wu et al. "Learning to See Physics via Visual De-animation". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/4c56ff4ce4aaf9573aa5dff913df997a-Paper.pdf.

[86] Michael Chang et al. "A Compositional Object-Based Approach to Learning Physical Dynamics". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Bkab5dqxe.

[87] David Zheng et al. *Unsupervised Learning of Latent Physical Properties Using Perception-Prediction Networks*. 2018. URL: https://arxiv.org/abs/1807.09244.

[88] Filipe de Avila Belbute-Peres et al. "End-to-End Differentiable Physics for Learning and Control". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/842424a1d0595b76ec4fa03c46e8d755-Paper.pdf.

[89] Michael Janner et al. "Reasoning About Physical Interactions with Object-Oriented Prediction and Planning". In: *International Conference on Learning Representations*. 2019.

[90] Fabien Baradel et al. "CoPhy: Counterfactual Learning of Physical Dynamics". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkeyppEFvS.

[91] Kevin Smith et al. "Modeling Expectation Violation in Intuitive Physics with Coarse Probabilistic Object Representations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/e88f243bf341ded9b4ced444795c3f17-Paper.pdf.

[92] Ronan Riochet et al. *IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning*. 2018. URL: https://arxiv.org/abs/1803.07616.

[93] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical Reparameterization with Gumbel-Softmax". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=rkE3y85ee.

[94] Christopher Nowzohour and Peter Bühlmann. "Score-based causal learning in additive noise models". In: *Statistics* 50.3 (2016), pp. 471–485.

[95] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. en. Ed. by Francis Bach. Adaptive Computation and Machine Learning series. Cambridge, MA, USA: MIT Press, Nov. 2017.

[96] Peter Spirtes and Clark Glymour. "An Algorithm for Fast Recovery of Sparse Causal Graphs". en. In: *Social Science Computer Review* 9.1 (Apr. 1991). Publisher: SAGE Publications Inc, pp. 62–72. URL: https://doi.org/10.1177/089443939100900106 (visited on 06/12/2022).

[97] Phillip Lippe et al. "iCITRIS: Causal Representation Learning for Instantaneous Temporal Effects". In: *UAI 2022 Workshop on Causal Representation Learning*. 2022. URL: https://openreview.net/forum?id=xeDKTZsZ7Z7.

[98] Aapo Hyvarinen and Hiroshi Morioka. "Unsupervised Feature Extraction by Time-Contrastive Learning and Nonlinear ICA". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/d305281faf947ca7acade9ad5c8c818c-Paper.pdf.

[99] Aapo Hyvarinen and Hiroshi Morioka. "Nonlinear ICA of Temporally Dependent Stationary Sources". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 20–22 Apr 2017, pp. 460–469. URL: https://proceedings.mlr.press/v54/hyvarinen17a.html.

[100] David A. Klindt et al. "Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=EbIDjBynYJ8.

[101] Ilyes Khemakhem et al. "Variational Autoencoders and Nonlinear ICA: A Unifying Framework". English. In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR*. Ed. by S Chiappa and R Calandra. Proceedings of Machine Learning Research. International Conference on Artificial Intelligence and Statistics , AISTATS ; Conference date: 26-08-2020 Through 28-08-2020. United States: ADDISON-WESLEY PUBL CO, 2020, pp. 2207–2216.

[102] Ilyes Khemakhem et al. "ICE-BeeM: Identifiable Conditional Energy-Based Deep Models". English. In: *NeurIPS2020*. Advances in Neural Information Processing. Conference on Neural Information Processing Systems (NeurIPS 2020), NeurIPS 2020 ; Conference date: 06-12-2020 Through 12-12-2020. United States: Neural Information Processing Systems Foundation, Dec. 2020.

[103] Roland S. Zimmermann et al. "Contrastive Learning Inverts the Data Generating Process". In: Proceedings of Machine Learning Research 139 (2021), pp. 12979–12990. URL: http://proceedings.mlr.press/v139/zimmermann21a.html.

[104] Luigi Gresele et al. "The Incomplete Rosetta Stone problem: Identifiability results for Multi-view Nonlinear ICA". In: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Ed. by Ryan P. Adams and Vibhav Gogate. Vol. 115. Proceedings of Machine Learning Research. PMLR, 22–25 Jul 2020, pp. 217–227. URL: https://proceedings.mlr.press/v115/gresele20a.html.

[105] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[106] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *ICLR (Poster)*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

[107] Aaron Golden et al. "High Speed Phase-Resolved 2-d UBV Photometry of the Crab pulsar". In: *Astronomy and Astrophysics* 363 (Nov. 2000).

[108] Andrew Lyne and Francis Graham-Smith. "9 The Crab and Vela Pulsars". In: *Pulsar Astronomy*. 4th ed. Cambridge Astrophysics. Cambridge University Press, 2012, pp. 117–133.

[109] Thomas Schlegl et al. “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery”. In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 146–157.

[110] Abdelkader Dairi et al. “Unsupervised obstacle detection in driving environments using deep-learning-based stereovision”. In: *Robotics and Autonomous Systems* 100 (2018), pp. 287–301. URL: https://www.sciencedirect.com/science/article/pii/S0921889017304736.

[111] Justin Johnson et al. “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2901–2910.

[112] Tomas Jakab et al. “Unsupervised Learning of Object Landmarks through Conditional Image Generation”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/1f36c15d6a3d18d52e8d493bc8187cb9-Paper.pdf.

[113] Phillip Lippe et al. “BISCUIT: Causal Representation Learning from Binary Interactions”. In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Ed. by Robin J. Evans and Ilya Shpitser. Vol. 216. Proceedings of Machine Learning Research. PMLR, 31 Jul–04 Aug 2023, pp. 1263–1273. URL: https://proceedings.mlr.press/v216/lippe23a.html.

[114] Mengyue Yang et al. “CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models”. en. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 9588–9597. URL: https://ieeexplore.ieee.org/document/9578520/ (visited on 06/08/2022).

[115] Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs: Theory, Algorithms, and Applications*. Springer London, 2009.

Appendices

A

Deep Learning of Sparse Causal Graphs using Additive Noise Model Priors

A.1 Proof of DAG maximality

Proposition A.1.1. *The set of all SEMs* $\mathcal{S} = \{(z_i = f_i(z_{pa_G(i)}))_{i=1}^d \,|\, \forall G, f\}$ *with arbitrary DAGs* G *and probabilistic functions* f *is contained within the set* $\mathcal{S}^\Omega = \{(z_i = f_i(z_{1,\ldots,i-1}))_{i=1}^d \,|\, \forall f\}$*, up to reorderings of the variables* z*.*

Proof. Because G is a DAG it has an acyclic ordering of its vertices [115]; i.e. if G has vertices $V = \{v_1, ..., v_n\}$ we can always construct a sequence $(z_1, ..., z_n)$ such that the set $Z = \{z_1, ..., z_n\}$ is mapped one-to-one to the set V, and such that z_i is an ancestor of z_j (i.e. there exists a path from z_i to z_j in G) for all $i < j$. Therefore, the set of parents $\mathrm{pa}_G(z_j)$ has to be a subset of the set $\{z_i | i < j\}$, from which Proposition A.1.1 follows. □

B

Identifiable Representation Learning from Architecture Equivariances

B.1 Proof of Theorem 5.3.1

Theorem (Maximum Position Error). Consider a set of images $x \sim X$ with objects of size s_o, CNN encoder ψ with receptive field size s_ψ, CNN decoder ϕ with receptive field size s_ϕ, soft argmax function softargmax, rendering function render with Gaussian standard deviation σ_G and $\Delta_G \sim \mathcal{N}(0, \sigma_G^2)$, and latent variables z, composed as $z = \text{softargmax} \circ \psi \circ x$ and $\hat{x} = \phi \circ \text{render} \circ z$ (fig. 5.1). Assuming (1) the objects are reconstructed at the same positions as in the original images, (2) each object appears in at least two different positions in the dataset, and (3) there are no two identical objects in any image, then the learned latent variables z correspond to the true object positions up to object permutations and maximum position errors Δ of

$$\Delta = \min\left(\frac{s_\psi}{2} + \frac{s_o}{2} - 1, \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G\right). \tag{B.1}$$

Proof. By assumption (1), the positions (z_1, z_2) objects in the original image x have to be the same as the positions $(\hat{z}_1, \hat{z}_2)$ of the objects in the reconstructed image $\hat{x}$. In practice, this occurs whenever the reconstruction loss is minimised.

By assumption (2) (each object appears at a minimum of 2 different positions), the latent variables used by the decoder have to contain some information about each object, and thus the encoder has to learn to match all the objects. This is because the decoder CNN ϕ takes as its input the rendered Gaussians $\hat{e} = \text{render} \circ z$ concatenated with positional encodings or a randomly sampled nearby frame (fig. 5.1), and if some object in the dataset only appeared at a single position the model could achieve perfect reconstruction solely by using the positional encodings or the conditioned image without having to use the Gaussian maps $\hat{e}$. However, because the dataset contains each object at a minimum of 2 positions, relying purely on positional encodings or on the conditioned image is now not sufficient, as without any information about the object passed to the decoder, it would be impossible for it to predict where to render the object. More formally, because $\hat{x} = \phi \circ \text{render} \circ \text{softargmax} \circ \psi \circ x$, this means that for objects in x to be reconstructed at the same positions as in $\hat{x}$ (assumption 1), the encoder ψ needs to match some part of each object in x.

Because $z = \text{softargmax} \circ \psi \circ x$ is equivariant to translations of x (sec. 5.2.2), and because the encoder ψ has to match some part of each object in x (as shown previously), and also each image consists of distinct objects (assumption 3) on a known background, the image x with an object at the position (u_1, u_2) is encoded by $\text{softargmax} \circ \psi$ to the latent variables

$$(z_1, z_2) = (u_1 + \Delta_{\psi 1}, u_2 + \Delta_{\psi 2}), \quad |\Delta_{\psi 1}|, |\Delta_{\psi 2}| \leq \frac{s_\psi}{2} + \frac{s_o}{2} - 1 \tag{B.2}$$

where the shifts $\Delta_{\psi 1}$ and $\Delta_{\psi 2}$ arise because any part of the encoder filter (of size s_ψ) can match any part of the object (of size s_o). See fig. 5.2a for an illustration.

Next, because $\hat{x} = \phi \circ \text{render} \circ z$ is equivariant to translations of z (sec. 5.2.3), the latent variables $z = (u_1 + \Delta_{\psi 1}, u_2 + \Delta_{\psi 2})$ are mapped to a predicted image $\hat{x} = \phi \circ \text{render} \circ z$ with an object at position

$$(\hat{z}_1, \hat{z}_2) = (u_1 + \Delta_{\psi 1} + \Delta_{\phi 1}, u_2 + \Delta_{\psi 2} + \Delta_{\phi 2}), \quad |\Delta_{\phi 1}|, |\Delta_{\phi 2}| \leq \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G \tag{B.3}$$

where the shifts $\Delta_{\phi 1}$ and $\Delta_{\phi 2}$ arise because any part of the decoder filter (of size s_ϕ) can match any part of the the rendered Gaussian $\hat{e}_t = \text{render} \circ z$, where $\Delta_G \sim \mathcal{N}(0, \sigma_G^2)$. See fig. 5.2b for illustration.

Finally, by assumption (1), the position of each object in the original image x has to be equal to the position of the object in the reconstructed image, i.e.

$$(u_1, u_2) = (u_1 + \Delta_{\psi 1} + \Delta_{\phi 1}, u_2 + \Delta_{\psi 2} + \Delta_{\phi 2}) \tag{B.4}$$

This results in the conditions

$$\Delta_{\psi 1} + \Delta_{\phi 1} = 0, \quad \Delta_{\psi 2} + \Delta_{\phi 2} = 0 \tag{B.5}$$

and therefore

$$|\Delta_{\psi 1}| = |\Delta_{\phi 1}|, \quad |\Delta_{\psi 2}| = |\Delta_{\phi 2}| \tag{B.6}$$

In words, the shift in the latent variables acquired from the encoder Δ_ψ has to be balanced by an opposite shift of the same magnitude in the decoder Δ_ϕ in order to reconstruct the object at the same position. Because the shift due to the encoder is of maximum magnitude of $s_\psi/2 + s_o/2 - 1$ and the shift due to the decoder has a maximum magnitude of $s_\phi/2 - s_o/2 + \Delta_G$, this means that the maximum magnitude of the shift of the latent variables has to be the minimum of these two expressions, i.e. the learned latent variables (z_1, z_2) correspond to the ground truth latent variables (u_1, u_2) up to

$$(z_1, z_2) = (u_1 + \Delta_1, u_2 + \Delta_2), \quad |\Delta_1|, |\Delta_2| \leq \min\left(\frac{s_\psi}{2} + \frac{s_o}{2} - 1, \frac{s_\phi}{2} - \frac{s_o}{2} + \Delta_G\right). \tag{B.7}$$

Additionally, because the order in which the objects get mapped to each latent variable is arbitrary, there is an additional indeterminacy arising due to variable permutations. □

B.2 Theoretical Results for Multiple Object Sizes

The results of corollary 5.3.2 can be extended to objects with a range of different sizes. The bound can be obtained by taking the maximum over all the bounds for objects with sizes $s_o \in [s_o^{min}, s_o^{max}]$, leading to the following corollary.

Corollary B.2.1 (Error vs. Encoder RF Size for Multiple Object Sizes)**.** *The maximum position error as a function of the encoder receptive field (RF) size s_ψ for a given s_ϕ, $s_o \in [s_o^{min}, s_o^{max}]$, σ_G, is*

$$\Delta(s_\psi) = \begin{cases} \frac{s_\psi}{2} + \frac{s_o^{max}}{2} - 1 & \text{if } \ 1 \leq s_\psi \leq s_\phi - 2s_o^{max} + 2, \\ \frac{s_\psi}{4} + \frac{s_\phi}{4} - \frac{1}{2} + \Delta_G & \text{if } \ s_\phi - 2s_o^{max} + 2 < s_\psi \leq s_\phi - 2s_o^{min} + 2, \\ \frac{s_\phi}{2} - \frac{s_o^{min}}{2} + \Delta_G & \text{if } \ s_\psi > s_\phi - 2s_o^{min} + 2. \end{cases}$$

For an illustration see fig. B.1a.

Similar to corollary 5.3.2, the results of corollary 5.3.3 can be extended to objects with a range of different sizes by taking the maximum over the bounds for objects with sizes $s_o \in [s_o^{min}, s_o^{max}]$, leading to the following corollary.

Corollary B.2.2 (Error vs. Decoder RF Size for Multiple Object Sizes)**.** *The maximum position error as a function of the decoder receptive field (RF) size s_ϕ for a given s_ψ, $s_o \in [s_o^{min}, s_o^{max}]$, σ_G, is*

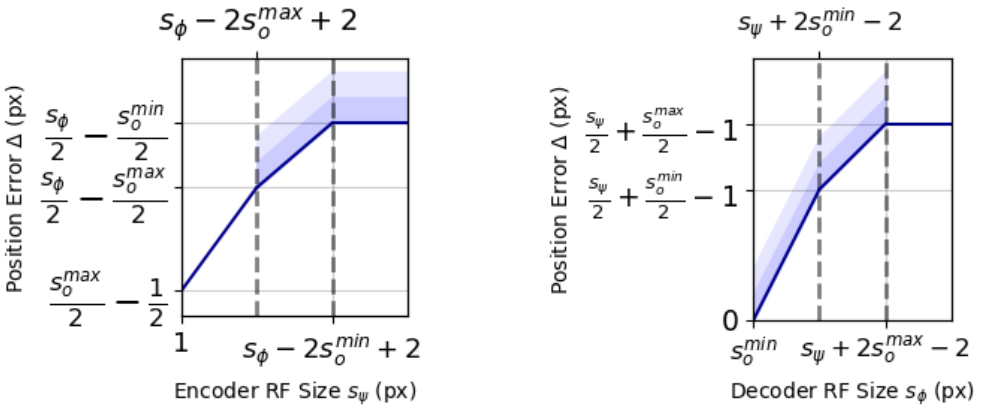
$$\Delta(s_\phi) = \begin{cases} \frac{s_\phi}{2} - \frac{s_o^{min}}{2} + \Delta_G & \text{if } \ s_o^{min} \leq s_\phi \leq s_\psi + 2s_o^{min} - 2, \\ \frac{s_\psi}{4} + \frac{s_\phi}{4} - \frac{1}{2} + \Delta_G & \text{if } \ s_\psi + 2s_o^{min} - 2 < s_\phi \leq s_\psi + 2s_o^{max} - 2 \\ \frac{s_\psi}{2} + \frac{s_o^{max}}{2} - 1 & \text{if } \ s_\phi > s_\psi + 2s_o^{max} - 2. \end{cases}$$

For an illustration see fig. B.1b.

B.3 Experiment Training Details

B.3.1 Synthetic Experiments

Dataset. In all of the synthetic experiments, we use the following setup. The dataset consists of black images of size $s_{img} \times s_{img} = 80 \times 80$ px, each with a white square with dimensions $s_o \times s_o$ px, centered at positions (x, y) where $x, y \in \{s_{pad} + s_o/2, s_{pad} + s_o/2 + 1, \ldots, s_{img} - s_{pad} - s_o/2\}$, where $s_{pad} \geq \max(s_\psi, s_\phi) - 1$ (to prevent unwanted edge effects). We divide the dataset into 4 quadrants and assign images from 3 quadrants to the training set and the remaining quadrant to the test set.



(a) Position error vs. encoder RF size for a range of object sizes.

(b) Position error vs. decoder RF size for a range of object sizes.

Figure B.1: Theoretical bounds for the maximum position error as a function of the encoder receptive field size s_ψ and the decoder receptive field size s_ϕ for objects of sizes ranging from s_o^{min} to s_o^{max}, as the remaining factors are kept constant. Each theoretical bound consists of two regions separated by a dashed line, one where the maximum error is due to the encoder (deterministic, represented by a solid line), and one where the maximum error is due to the decoder (probabilistic, represented by a solid line and shaded regions). Areas within one and two standard deviations of the mean are represented by a darker and lighter shades of blue respectively.

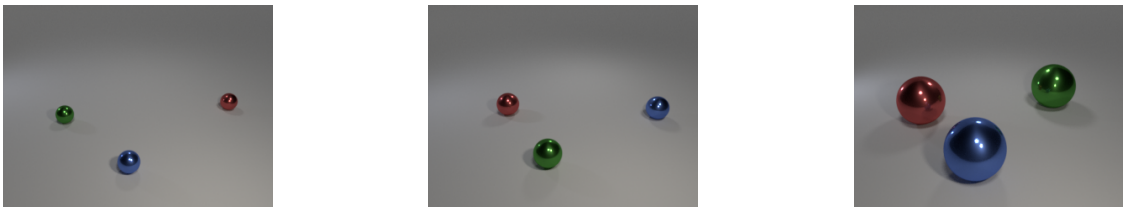
Evaluation. For each experiment, we fix all but one variable from the following set: encoder RF size s_ψ, decoder RF size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, and vary the remaining variable. For each value of the investigated variable we perform 20 experiments with different random seeds, noting down the learned position error Δ as the absolute difference between the learned position z and the centre of the object z_{GT} (maximum over horizontal and vertical differences, and over the test set). We discard a result if the reconstruction accuracy of the run is below 99.9%, to only consider runs where the object has been detected successfully (this is because the square is on the order of 7×7 px, thus only comprising 0.8% of the image).

Architecture. We parametrise both the encoder ψ and the decoder ϕ as 5-layer CNNs with Batch Normalisation and ReLU activations, 32 channels, and filter sizes in $\{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$, such that their receptive field sizes equal s_ψ and s_ϕ respectively. We train each network for 500 epochs using the Adam optimiser with

learning rate 10^{-3} and batch size 128. We train each experiment on a single GPU for around 6 hours with <6GB memory on an internal cluster.

B.3.2 CLEVR Experiments

Dataset. Our CLEVR experiments use data generated with the CLEVR [111] image generation script. Our training and test sets consist of 150 and 50 images respectively, containing red, green and blue metallic spheres on a random background, at random positions, and with a different range of sizes (fig. B.2). For experiments measuring position error as a function of encoder and decoder RF sizes and Gaussian s.d., we use a dataset with object sizes between 6-10 px after perspective distortion (fig. B.2a). For the experiment measuring position error as a function of object size, we use 5 datasets with object sizes 9-14 px, 11-17 px, 13-19 px, 15-24 px, 17-27 px after perspective distortion (figs. B.2b, B.2c).



(a) Object sizes 6-10 px.

(b) Object sizes 9-14 px.

(c) Object sizes 17-27 px.

Figure B.2: Samples from our CLEVR datasets, with object sizes (a) 6-10 px, (b) 9-14 px, (c) 17-27 px.

Evaluation. For each experiment, we fix all but one variable from the following set: encoder RF size s_ψ, decoder RF size s_ϕ, object size s_o, and Gaussian standard deviation σ_G, and vary the remaining variable. For each value of the investigated variable we perform experiments with different random seeds, noting down the learned position errors Δ as the absolute difference between the learned position z of each object and the centre of the object z_{GT} (maximum over horizontal and vertical differences and over the test set, for each object). We only consider results for objects that have been learned successfully. For experiments measuring position error as a function of encoder and decoder RF sizes and Gaussian s.d., we consider

an object to be learned if the position error is less than 35 px, and if only a single variable corresponds to the object, and if the position error is stable over consecutive training iterations. For the experiment measuring position error as a function of object size, we only consider runs where all 3 objects have been learned, i.e. where the reconstruction accuracy is higher than 98.0% for the dataset with object sizes 6-10 px, 98.8% for dataset with sizes 11-17 px, 98.0% for dataset with sizes 13-19 px, 97.5% for dataset with sizes 15-24 px, and 98.0% for dataset with sizes 17-27 px.

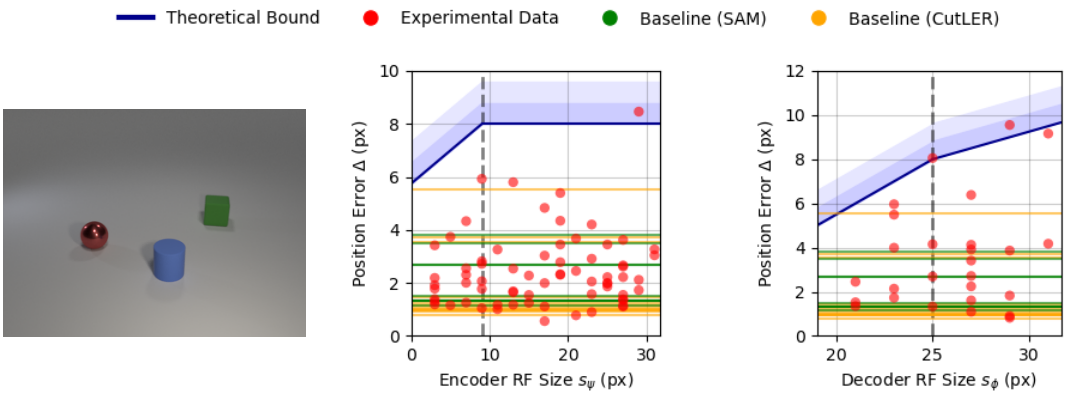
Architecture. We parametrise both the encoder ψ and the decoder ϕ as 5-layer CNNs with Batch Normalisation and ReLU activations, 32 channels, and filter sizes in $\{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$, such that their receptive field sizes equal s_ψ and s_ϕ respectively. We train each network until convergence using the Adam optimiser with learning rate 10^{-2} and batch size 128 for the experiments measuring position error as a function of encoder and decoder RF size and Gaussian s.d., and with learning rate 10^{-3} and batch size 150 for the experiment measuring position error as a function of object size. We train each model on a single GPU for less than day with <8GB memory on an internal cluster.

Baselines. We also evaluate the results for two State-of-the-Art baselines, SAM [11] and CutLER [12]. First, for each of our 6 datasets (containing objects of sizes 6-10 px, ..., 17-27 px), we combine its training and test set to create 4 sets of 50 images each. We then apply SAM and CutLER to all images in each 50-image set, noting down the learned position errors Δ as the absolute difference between the predicted position z of each object and the centre of the object z_{GT} (maximum over horizontal and vertical differences and over the 50-image set, for each object). For SAM, we take the predicted object positions to be the centres of the bounding boxes corresponding to the second, third and fourth predicted masks (first mask corresponding to the background). For CutLER, we take the predicted object positions to be the centres of the predicted bounding boxes if the method predicts 3 bounding boxes (one for each object), otherwise we discard the prediction. For both methods, we discard any result where position error is greater than 35 px, to

be consistent with the evaluation for our method. Finally, this results in 12 position error values (3 objects × 4 data splits), for each method and each of our 6 datasets.

B.4 CLEVR Experiments with Different Shapes

To demonstrate that our method applies to objects of any shape, in this section we include experiments on our CLEVR dataset with three distinct objects – a red metallic sphere, a blue rubber cylinder and a green rubber cube. The dataset has objects of size 9-19 px after perspective distortion, and a sample from the dataset is shown in fig. B.3a. We perform training and evaluation in the same way as in app. B.3.2. We plot the position error as a function of the encoder and decoder receptive field sizes in figs. B.3b and B.3c, respectively. We can observe that all our experimental data (red) lies within the bounds predicted by our theory (blue), successfully validating our theory for objects with different shapes.



(a) Dataset Sample. **(b)** Error vs. Encoder RF. **(c)** Error vs. Decoder RF.

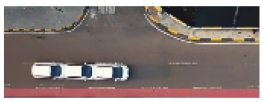
Figure B.3: CLEVR experiment results using (a) a dataset with 3 objects of different shapes (a sphere, a cube, and a cylinder), showing position error as a function of (b) the encoder receptive field size s_ψ and (c) decoder receptive field size s_ϕ, as the remaining factors are fixed to $s_\psi = 9, s_\phi = 25$, object size $s_o \in [9, 19]$ and Gaussian s.d. $\sigma_G = 0.8$. Theoretical bounds are denoted by blue, experimental results in red, SAM baseline in green, and CutLER baseline in orange.

B.5 Experiments with Real Videos

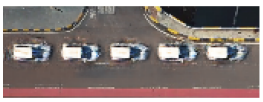
In this section we present experimental results of applying our method to real YouTube videos, including overhead traffic footage and mini pool game footage.

B.5.1 Traffic Data

In this experiment we aim to learn the position of a car from an overhead traffic video. The training and test sets for this experiment consist of 25 frames each, from a video of an overhead view of a car moving for a short distance in a single lane (fig. B.4a). We train the architecture from fig. 5.1 on this training set and validate it on the test set. It achieves a mean squared error between the ground truth object positions and the learned object positions of $7.2 \cdot 10^{-5}$ (in units normalised by the image size), demonstrating that the object position has been learned successfully with a very low error. We then modify the learned position variable and decode it to generate videos of the car at novel positions (figs. B.4b, B.4c).



(a) Training data.



(b) Generated data (steady speed in a different lane).



(c) Generated data (lane change and acceleration).

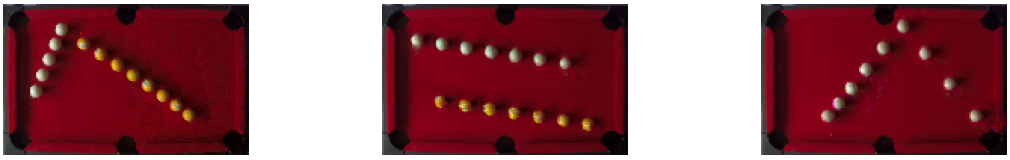
Figure B.4: Road traffic video used for training, together with two videos generated after training by modifying and decoding the learned latent variables (video frames are superimposed). The car object is detected successfully and is used to generate realistic videos with objects at unseen positions.

B.5.2 Mini Pool

In this experiment we aim to learn the positions of balls from a video of a game of mini pool. The training and test sets for this experiment consist of 15 and 11 frames respectively from a video of a game of mini pool, cropped to a portion

where two balls are moving at the same time (fig. B.5a). We train the architecture from fig. 5.1 on this training set and validate it on the test set, where it achieves a mean squared error between the ground truth object positions and the learned object positions of $8.2 \cdot 10^{-3}$ (in normalised units). This demonstrates that the object positions were learned successfully with a very low error. We then modify the learned position variables and decode them to generate videos of the balls at novel positions (figs. B.5b, B.5c).

In practice, it was important to set the encoder and decoder receptive field sizes to be greater than but close to the size of the objects, as for larger RF sizes the position error increased unnecessarily and the images were rendered further away from the position given by the latent variables. Also, for large receptive field sizes, the decoder filter contained too much background which caused low quality reconstructions when rendering the balls near the mini pool table edges.



(a) Training data.

(b) Generated data (linear motion at unseen positions).

(c) Generated data (collision and slowing down).

Figure B.5: Mini pool video used for training, together with two videos generated after training by modifying and decoding the learned latent variables (video frames are superimposed). Both ball objects are detected successfully and are used to generate realistic videos.

C

Sources of Figures

- "Text generation" in fig. 1.1a: by Marian Longa, source from https://chat.openai.com.
- "Image generation" in fig. 1.1b: by Marian Longa, source from https://chat.openai.com.
- "Protein prediction" in fig. 1.1c: by EMBL-EBI, source from https://alphafold.ebi.ac.uk/entry/H2NHM8.
- "Playing chess" in fig. 1.1d: by Marian Longa, source from https://chat.openai.com.
- "Explainability" in fig. 1.2a: by Dominic Alves, source from https://www.flickr.com/photos/dominicspics/34767819644, under the CC BY 2.0 DEED Licence.
- "Generalisation" in fig. 1.2b: by savagecat, source from https://www.flickr.com/photos/catrionasavage/4184248150, under the CC BY 2.0 DEED Licence.

- "Learning guarantees" in fig. 1.2c: by Eschenzweig, source from https://en.wikipedia.org/wiki/File:Autonomous-driving-Barcelona.jpg, under the CC BY-SA 2.0 DEED Licence.
- "Robustness" in fig. 1.2d: by John Cunniff, source from https://flickr.com/photos/131474603@N03/21920624861, under the CC BY 2.0 DEED Licence.
- "Structural causal model" in fig. 1.3a: by Marian Longa.
- "Causal graph" in fig. 1.3b: by Marian Longa.