

Combine and Conquer:

Representation Learning from Multimodal Data



Yuge Shi (Jimmy)

St Hilda's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2023

Acknowledgements

I am deeply, deeply grateful to my amazing advisor, Phil Torr, who has been a great source of support and fun. His open-mindedness has allowed me to explore my own path, and his inquisitive mind have been instrumental in shaping my research. I am also in awe of his words of wit and wisdom, which continue to guide me in all aspects of my life. I must also express my gratitude to my other advisor, Siddharth N., who has shown endless patience and generosity, especially during the early stages of my PhD. Not only did teach me everything I know about generative models, but his rigorous, scientific approach to thinking and his ability to communicate clearly and concisely have had a profound impact on me and are qualities I strive to develop going forward. I feel incredibly lucky to have such wonderful advisors who supported me unconditionally through difficult times, and who championed me and my work more than I could ever expect.

I was also very fortunate to have had the opportunity to work with Gabriel Synnaeve at FAIR during my internship, who believed in me even when I doubted myself. To be honest I don't really understand what I could have possibly done to have earned this level of trust and support I felt from him — one can only attribute this to his generosity and exceptional level of empathy — and I cannot put in words just how much this encouraged and inspired me. I have learned so much from him and other researchers from his team, including Nicolas Usunier, Awni Hannun and Jeffrey Seely. I would like to express my gratitude to my advisor from DeepMind, Adam Kosior, for offering me the opportunity to work on the ADIOS project, which was such an enjoyable and rewarding experience. As a supervisor, Adam strikes the perfect balance between being hands-on and hands-off, offering guidance when needed but also encouraging me to explore my own path. I have benefited greatly from his approach and I aspire to be a similar type of supervisor for my own students in the future.

My PhD would have been much less enjoyable without the fellow students who I have had the opportunity to work with, including (in chronological order) Amartya Sanyal, Tom Joy and Imant Daunhawer. What I learned from working with them is something I will carry with me for my entire career, for which I am so grateful. I am also incredibly appreciative of the Oxford Women in Computer Science (OxWoCS)

society, for being a constant source of friendship, empowerment and support. My deepest gratitude goes to the amazing people I met through there who I look up to so much, including but not limited to Catherine Tong, Klaudia Krawiecka, Linde Hesse and Maddy Wyburd.

My partner in crime (and life) Ben has been instrumental in my survival throughout the PhD — I think you deserve half of my degree. Thank you for moving from the wonderful, sunny, everything-is-chill Australia, to, well, here, and for sharing my highest high and lowest low. I am so fortunate to share part of my life with you.

I would also like to thank my parents, despite the fact that they won't understand any of this because they don't speak English, for raising me to become the person that I am today. As I grow older I see more and more of my parents in myself, and I feel lucky to say that I am proud of that.

This is my 8th year abroad, and the petrified 19-year-old me who had just left China would have been so happy to know the amazing friends I got to make during my PhD. In rough chronological order, thank you to Jingyue Lu, Elspeth Windsor, Nantas Nardelli, Csabi Botos, Joanna Materzynska, Jishnu Mukhoti, Amartya Sanyal, Tom Joy, Catherine Tong, Linde Hesse, Maddy Wyburd, Klaudia Krawiecka, Pau de Jorge, Shuyang Sun, Chen Lin and Francisco Girbal Eiras. A special thank you to the amazing Elise van der Pol who has been like a big sister to me, Shati Patel and Sofia Vaz-Pinto-Simoes-Coelho for all the fun lead climbing times, and to my dog Biang Biang for being fluffy and providing much needed cuteness therapy.

Finally, a huge thank you to everyone who has been part of this journey. It's been real.

Abstract

Supervised learning, which involves training a model using a labelled dataset, is becoming less popular due to its high cost and various issues with generalization and robustness. This is unsurprising, as data such as images and language are complex and cannot be accurately represented by a single label. When models are trained using this method, they often learn features that spuriously correlate with the label, resulting in poor performance when deployed in the real world.

This thesis explores representation learning using multiple sources of data, such as images and language or photos and sketches. We demonstrate through both generative and discriminative models that extracting common abstract concepts between multiple modalities or domains can lead to more accurate and generalisable representations. In addition, we investigate ways to improve the data efficiency of these models, including using fewer multimodal pairs through contrastive-style objectives and generating multimodal pairs through masked image modeling. Finally, we systematically evaluate the robustness of different learning objectives on distribution shift tasks in order to understand their usefulness in the real world.

Contents

Bibliography Notes	xii
1 Introduction	1
2 Background	5
2.1 Variational Autoencoder	5
2.1.1 Understanding VAE	6
2.1.2 Variational inference	8
2.1.3 Deriving Evidence Lower Bound (ELBO)	9
2.1.4 Deriving Importance Weighted Autoencoder (IWAE)	10
2.1.5 Gradient variance and its reduction	12
2.2 Multimodal Deep Generative Models	15
2.2.1 Multimodal VAEs	15
2.2.2 Large-scale multimodal generative models	16
2.3 Domain Generalisation	18
2.3.1 Introduction	18
2.3.2 Literature review	19
2.4 Augmentation-based Self-supervised Learning	21
2.4.1 Introduction	22
2.4.2 Literature review	23

3	Variational Mixture-of-Experts Autoencoders for Multimodal Deep Generative Models	26
3.1	Introduction	28
3.2	Related Work	30
3.3	Methods	32
3.4	Experiments	35
3.4.1	Common Details	36
3.4.2	MNIST-SVHN	36
3.4.3	CUB Image-Captions	40
3.5	Discussions	43
4	Relating by Contrasting: A Data-efficient Framework for Multimodal DGMs	45
4.1	Introduction	47
4.2	Related Work	48
4.3	Methodology	51
4.3.1	Contrastive loss for “relatedness” learning	52
4.3.2	Optimising the objective	54
4.4	Experiments	56
4.4.1	Datasets	56
4.4.2	Metrics	56
4.4.3	Improved multimodal learning	58
4.4.4	Data Efficiency	60
4.4.5	Label propagation	61
4.5	Conclusion	63

5	Gradient Matching for Domain Generalization	65
5.1	Introduction	67
5.2	Related Work	69
5.3	Methodology	72
5.3.1	Goals	72
5.3.2	The pitfall of ERM: a linear example	72
5.3.3	Inter-domain Gradient Matching (IDGM)	74
5.3.4	Optimizing IDGM with Fish	75
5.4	Experiments	78
5.4.1	CDSPRITES-N	78
5.4.2	WILDS	79
5.4.3	DOMAINBED	81
5.4.4	Analysis	82
5.5	Conclusion	85
6	Adversarial Masking for Self-Supervised Learning	86
6.1	Introduction	88
6.2	Methodology	90
6.2.1	Inference model $\mathcal{I}$	90
6.2.2	Occlusion model $\mathcal{M}$	93
6.2.3	Putting it together	94
6.3	Evaluation of Representations	96
6.3.1	Classification	96
6.3.2	Transfer learning	98
6.3.3	Robustness	100
6.4	Analysis on Learned Masks	101
6.4.1	Mask Generation	102
6.4.2	Comparing Masking Schemes	103

6.5	Related Work	105
6.6	Conclusion	107
7	Epilogue	109
7.1	Summary and impact	110
7.2	Limitations	112
7.3	Future works	113
	Bibliography	116
	Appendices	
A	How Robust is Unsupervised Representation Learning to Distribu- tion Shift?	156
A.1	Introduction	158
A.1.1	Existing problems and Contributions	159
A.2	Setting Up	163
A.3	Experimental results	166
A.3.1	Synthetic distribution shift	166
A.3.2	Real-world distribution shift	170
A.4	Related Work	174
A.5	Conclusion and future work	175
B	Appendix: Variational Mixture-of-Experts Autoencoders for Mul- timodal Deep Generative Models	177
B.1	Tighter lower bound	177
B.2	Multimodal Importance-Sampled ELBO	179
B.3	The DReG Estimator	180
B.4	Higher entropy of Importance Weighted Autoencoder (IWAE) varia- tional posteriors	180
B.5	Qualitative results of MVAE implementation with MOE posterior	181

B.6	Encoder and decoder architectures	183
B.7	Qualitative results of MMVAE on CUB	184
B.8	Qualitative results of Wu and Goodman (2018a)’s MVAE on CUB	186
C	Appendix: Relating by Contrasting: A Data-efficient Framework for Multimodal DGM	188
C.1	From pointwise mutual information to joint marginal likelihood	188
C.2	Connection of final objective to Pointwise Mutual Information	189
C.3	Generalisation to $M > 2$ Modalities	190
C.4	The Ineffectiveness of training with $\mathcal{L}_C$ only	192
C.5	Ablation study of γ	193
C.6	Architecture	197
C.7	Qualitative Results on MNIST-SVHN	200
C.7.1	Generative Results & Marginal Likelihoods on MNIST-SVHN	200
C.7.2	Generation diversity	205
C.8	Qualitative results on CUB	206
D	Appendix: Gradient Matching for Domain Generalization	208
D.1	Taylor Expansion of Reptile and Fish Inner-loop Update	208
D.1.1	Fish and Reptile: Differences and Connections	211
D.1.2	When $\epsilon = 1$	212
D.2	SmoothFish: a more general algorithm	214
D.2.1	Derivation	214
D.2.2	Results	214
D.3	CdSprites Architecture	216
D.4	Discussions and Results on WILDS	216
D.4.1	POVERTYMAP-WILDS	217
D.4.2	CAMELYON17-WILDS	217
D.4.3	FMoW-WILDS	219

D.4.4	CIVILCOMMENTS-WILDS	220
D.4.5	iWILDCAM-WILDS	221
D.4.6	AMAZON-WILDS	222
D.5	Hyperparameters	223
D.6	Ablation Studies on Pre-trained Models	223
D.7	Ablation Studies on hyperparameters	224
D.8	Tracking gradient inner product	225
D.9	Summary of DOMAINBED	227
D.10	Algorithm for tracking gradient inner product	227
D.11	t-SNE Visualisation of Learned Representations	227
E	Appendix: Adversarial Masking for Self-Supervised Learning	230
E.1	ADIOS Objectives	230
E.1.1	SimCLR	230
E.1.2	SimSiam	231
E.1.3	BYOL	231
E.2	Backbone of Occlusion Model	233
E.3	Setups of Classification Tasks	233
E.4	Setups of Transfer Learning	234
E.5	Setups of CLEVR	234
E.6	CIFAR ResNet	235
F	Appendix: How Robust is Unsupervised Representation Learning to Distribution Shift?	237
F.1	Additional Experimental Results	237
F.1.1	CdSprites	237
F.2	Architecture and Hyperparameters	238
F.2.1	MNIST-CIFAR	239
F.2.2	CdSprites	240

F.2.3	Camelyon17 and FMoW	241
F.3	Constructing Camelyon17-CS and FMoW-CS	243
F.4	Other Related Work	244

Bibliography Notes

The main text of this thesis is composed of four first-authored publications at top conferences. We also include one first-authored publication currently under review in the appendix.

Some parts of Chapter 2 are based on my own blog posts, namely *How I learned to stop worrying and write ELBO in a billion ways* (Shi, 2020) and *An incomplete and slightly outdated literature review on augmentation based self-supervise learning* (Shi, 2021).

Chapter 3 is based on Shi et al. (2019b), which is joint first-authored with my DPhil advisor Prof. Siddharth Narayanaswamy, co-authored with Dr. Brooks Paige and my other DPhil advisor Prof. Philip Torr, published in Advances in Neural Information Processing Systems (NEURIPS) 2019.

Chapter 4 features Shi et al. (2020) published in International Conference on Learning Representations (ICLR) 2021, and is also co-authored with Dr. Brooks Paige, Prof. Philip Torr and Prof. Siddharth Narayanaswamy.

The work in Chapter 5 is done during my internship at Facebook AI Research (FAIR) with Dr. Gabriel Synnaeve, who is one of the co-authors of this paper. Other co-authors include both of my DPhil advisor Prof. Philip Torr and Prof. Siddharth Narayanaswamy, as well as a few other members at FAIR including Jeffrey Seely, Dr. Awni Hannun, Dr. Nicolas Usunier. This work was published in ICLR 2022.

Chapter 6 is based on Shi et al. (2022c), which is done in collaboration with Dr. Adam Kosiorek at DeepMind. Co-authors include Prof. Philip Torr, Prof. Siddharth Narayanaswamy and Dr. Adam Kosiorek. It was published in International Conference in Machine Learning (ICML) 2022.

Additionally, we include Shi et al. (2022a) in Appendix A, a work in collaboration with Imant Daunhawer from ETH Zurich, his supervisor Prof. Julia Vogt, Dr. Amartya Sanyal as well as Prof. Phil Torr. This work is currently under review at a conference.

1

Introduction

Imagine if you were an extraterrestrial, tasked to grasp the concept of “dog” as a first step of your conquest of Earth. If your colleague, Groot, suggests that you do this by looking at a million images of dogs and associating them with label “0”, and another million images of non-dogs and associating them with label “1”, you would probably think that they are trying to sabotage your career. Instead, a more productive (and, let’s be honest, more fun) approach would be to visit Earth and interact with some dogs yourself — allowing you the opportunity to see, hear, smell and feel them. In fact, studies on Earth have shown that human learning in the real world involves the observation and interaction with multiple modalities or domains, such as perceiving the same object through visual observation, linguistic description, or physical interaction (Stein et al., 2009; Quiroga et al., 2009). Joint learning of these different modalities have been shown to benefit reasoning and understanding through expression across these modalities (Yildirim, 2014; Bauer and Johnson-Laird, 1993; Fan et al., 2018).

Despite these insights, most machine learning models are trained using the *unimodal, supervised* learning regime advocated by Groot, where labels are used to train the model to extract features. This regime is used in state-of-the-art models

for almost all computer vision tasks (Bhat et al., 2021; Zhai et al., 2022; Liu et al., 2022a; Li et al., 2022a): for image classification, the labels are index of the class that the image belongs to; for depth estimation, per-pixel distance to object in the scene is used to supervise the learning of the model. While this approach may be effective for some tasks, it has several potential drawbacks. First off, obtaining a sufficient amount of labeled data can be expensive and time-consuming. Additionally, these algorithms can be brittle and prone to overfitting, especially when dealing with high-dimensional and complex data. They can also struggle with unbalanced datasets, where the distribution of classes is uneven (Abd Elrahman and Abraham, 2013; Buda et al., 2018). Recent research has also revealed that models trained using supervised learning are vulnerable to adversarial attacks (Szegedy et al., 2013), susceptible to simplicity bias (Shah et al., 2020) and can have poor out-of-distribution (OOD) generalization performance (Koh et al., 2021; Gulrajani and Lopez-Paz, 2020).

Multimodal learning, which leverages the complex and nuanced relationships between different modalities, could be a potential solution to some of the challenges of supervised learning. In addition to the aforementioned human learning inspiration, multimodal learning allows the model to learn abstract concept through leveraging the complex and nuanced relationships between different modalities, allowing for a more comprehensive understanding of a given problem or task. This could potentially address some of the robustness issues that supervised learning from simple, potentially noisy and biased labels seem to suffer from; in addition, multimodal data is much easier to acquire and requires no human annotations, as one can simply leverage the temporal consistency across different perceptual modalities. For instance, a cooking instruction video on YouTube features three perceptual modalities including images (video), text (caption) and audio (sound track). One can even consider these modalities as labels for each other, with the difference being that these “labels” are more complex and can contain information that is not included in the input modality.

In this thesis, we investigate learning representation from multimodal data. As the goal is to explore the benefits of embedding data from multiple sources jointly, we relax the definition of “modalities” from traditional perceptual modalities such as vision and language, to different data distributions, for instance the sketch and realistic photo of a dog, or two randomly augmented views of the same image. This allows us to look at the joint embedding problem under versatile machine learning settings, from domain generalisation, self-supervised learning to visual-language generative models. We show with both generative and discriminative models the advantages of learning from multimodal data, both in terms of performance on popular benchmark to robustness properties such as generalisation; we were also the first to propose systematic desiderata for multimodal deep generative models. Additionally, inspired by the limited computational resources and suitable datasets available to us at the time, we also investigated improving the data efficiency of multimodal learning, by exploring 1) using less multimodal pairs strictly and 2) generating multimodal data. We now introduce how contributions on these different topics are organised in this thesis.

Summary of contributions and organisation Chapter 2 provides the necessary backgrounds required to understand the rest of the thesis, including introductions to the problems studied as well as a comprehensive literature review.

In Chapter 3, we investigate multimodal learning under a generative model setting, where we propose a novel multimodal VAE named MMVAE that learns high quality, balanced representations for all its input modalities. Importantly, our work is the first multimodal VAE that can perform cross-modal generation (e.g. language $\rightarrow$ vision, vision $\rightarrow$ language) at test time without auxiliary components. In addition, we also propose four desiderata that a multimodal VAE should satisfy, which is now widely adopted in the field of multimodal VAE.

Chapter 4 continues the line of work on multimodal VAE and looks at improving the data efficiency of such models in general. We propose a contrastive framework that can be used for any multimodal VAE, and show through experiments that our

framework greatly reduce the amount of labelled multimodal pairs needed to achieve the same level of performance; note that our adaptation of contrastive objective to multimodal learning predates CLIP (Radford et al., 2021) and was novel at the time of publishing.

In Chapter 5 we re-examine the problem of multimodal learning under a discriminative lens and study domain generalisation, where models must leverage commonalities between multiple source domains in order to generalise well to the unseen target domain. We propose a method called Fish that encourages the learning of such commonalities through maximising the inter-domain gradient inner product.

Finally, in Chapter 6 we look at self-supervised learning (SSL), the learning principle of which can be considered as “multimodal” in nature as the learning objective relies on preserving the commonality between two augmented views of the same image. We propose to mask out one of the augmented views using masks generated adversarially to the image encoder, allowing us to occlude complete semantic entities in the image, yielding better quality representations learned from the image encoder.

Notably, the works covered in the main text of this thesis looks at three different types of representation learning regimes: reconstruction-based generative models (VAE), supervised learning and SSL. Most evaluation of these models follow standard statistical assumption, where the train and test data are drawn from the same distribution. However, this assumption does not hold in deployment of machine learning models. Therefore, in Appendix A we investigate how methods in these three different regimes generalise to unseen distribution to gauge how they perform in the real world. With careful design of experiments and evaluation metrics, we made the surprising discovery that unsupervised methods, especially SSL, generalise better to unseen distributions especially when the distribution shift is more extreme. Since this work does not fit as much into the multimodal theme compared to the rest of the thesis, we exclude it from the main text and direct curious readers to the appendix.

2

Background

2.1 Variational Autoencoder

This section introduces the key ideas behind Variational Autoencoder (VAE) and provides backgrounds to methods used in Chapters 3 and 4.

VAE was first introduced in Kingma and Welling (2014a) as a scalable and efficient method for performing Stochastic Variational Inference (SVI) (Hoffman et al., 2013). Since then, VAEs have been widely used in computer vision for tasks such as data compression and generation. They have also been used in structured representation learning, where researchers have debated whether they can recover the true generative factors in observed data, a question that relates to concepts such as disentanglement and identifiability (Bengio et al., 2013; Mathieu et al., 2019b; Hyvarinen and Morioka, 2017; Lee, 1998). In the following, we will introduce the main concept behind VAE as well as various techniques that improve and facilitate its optimisation.

2.1.1 Understanding VAE

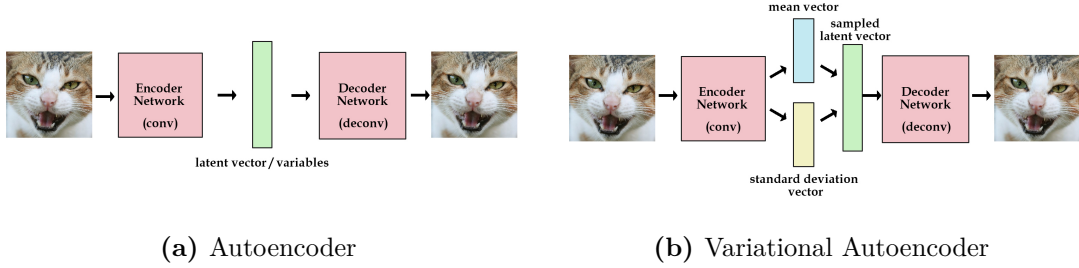


Figure 2.1: Pipeline of AE vs. VAE (image from (Frans, 2016))

An autoencoder perspective A popular way of conceptualising VAE is by considering it as an extension to Autoencoder (AE). As seen in Figure 2.1, both VAE and AE consists of two parts: an encoder, which maps an input data sample to a lower-dimensional latent space, and a decoder, which maps the latent representation back to the original data space. The only difference is that while AE generates the latent to a given observation directly as the output of the encoder, VAE samples its latent from a probabilistic distribution parametrised by the encoder. From this perspective, one can be tempted to simply regard VAE as an AE with a probabilistic latent space.

It is worth noting that while this viewpoint is intuitive and convenient, it is an oversimplification of VAE as it disregards the generative model assumption, which is important in understanding its ability to perform tasks such as data generation and density estimation. We will now provide an intuition to VAE from the generative model perspective.

A generative model perspective Let us assume that the observations we make in the real world, denote as $\mathbf{x}$, are all drawn from some underlying distribution $p(\mathbf{x})$. The goal is to train a model that learns a distribution $\hat{p}(\mathbf{x})$ that we can sample from, such that $\hat{p}(\mathbf{x})$ is as close to $p(\mathbf{x})$ as possible.

To simplify this task a bit, instead of generating each sample of $\mathbf{x}$ from scratch, one can imagine first defining some key characteristics of $\mathbf{x}$. For instance, if $\mathbf{x}$ is an

image of a dog, we can condition the generation of this image by information on the dog breed, activity depicted and the surroundings. This kind of specification is known as *latent variable*, which we denote as $\mathbf{z}$.

We also assume that the vector space $\mathcal{Z}$ where the probability distribution function (PDF) of $\mathbf{z}$ is defined in can be mapped to the space of observations $\mathcal{X}$ by a deterministic function f parametrised by θ , i.e. $f : \mathcal{Z} \times \Theta \rightarrow \mathcal{X}$. The goal is then to optimise deterministic function f such that $f_\theta(\mathbf{z})$ matches $p(\mathbf{x})$. To make the dependency of $\mathbf{x}$ on $\mathbf{z}$ explicit, we replace the notation $f_\theta(\mathbf{z})$ by $p_\theta(\mathbf{x}|\mathbf{z})$.

With these assumptions, we can now consider the inference of the model, i.e. infer values of latent variables $\mathbf{z}$ given observed data $\mathbf{x}$, using Bayes' Theorem:

$$p_\theta(\mathbf{z} | \mathbf{x}) = \frac{p_\theta(\mathbf{x} | \mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \quad (2.1)$$

Note that the probability of our observation, $p(\mathbf{x})$, appears in the denominator of Equation (2.1). We can calculate this probability by marginalising out $\mathbf{z}$, giving us

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \quad (2.2)$$

We can then optimise the generative model by maximising the probability of $p(\mathbf{x})$, where $\mathbf{x}$ is an observation from our dataset. The intuition here is, if the model deems training set samples likely, it is then equally likely to produce similar samples, indicating that the model has learned to match the distribution of observed data $p(\mathbf{x})$.

However, unfortunately Equation (2.2) is generally intractable, since the integration requires the evaluation over all possibilities of $\mathbf{z}$, which can be problematic when $\mathbf{z}$ is of high dimensionality. We consider two possible strategies to approximate the integral: Monte-Carlo (MC) method and importance sampling (IS).

Monte-Carlo (MC) method Approximate $p(\mathbf{x})$ by taking n samples of $\mathbf{z}$, i.e. $p_\theta(\mathbf{x}) \approx \frac{1}{N} \sum_N p(\mathbf{x} | \mathbf{z}^{(n)})$, $\mathbf{z}^{(n)} \sim p(\mathbf{z})$. However, the variance of the estimator can be big if $\mathbf{z}$ -space is large and we would potentially need millions of samples to get a reliable estimate;

Importance sampling (IS) Since sampling directly from $p(\mathbf{z})$ is problematic, we can resort to IS. IS allows us to, when evaluating the expectation of a *nominal* distribution, sample from an alternative *proposal* distribution that has more "important" values to reduce the variance of the estimator. It then weighs the simulation outputs by the probability ratio of the *nominal* distribution with respect to the *proposal* distribution to ensure that the estimation is unbiased. Using this technique, instead of sampling from $p(\mathbf{z})$, we can sample from a proposal posterior distribution $q_\phi(\mathbf{z} | \mathbf{x})$. One can imagine that using this proposal posterior, we can draw more "meaningful" samples from the latent space since they are conditioned on our observation x . We can write

$$\begin{aligned} p(\mathbf{x}) &= \int_{\mathbf{z}} p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})}[p_\theta(\mathbf{x} | \mathbf{z})] \\ &= \mathbb{E}_{p(\mathbf{z})} \left[\frac{q(\mathbf{z} | \mathbf{x})}{q(\mathbf{z} | \mathbf{x})} p_\theta(\mathbf{x} | \mathbf{z}) \right] = \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} \left[\frac{p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \right]. \end{aligned} \quad (2.3)$$

However, this comes with another problem: it turns out the optimal proposal distribution $q_\phi^*(\mathbf{z} | \mathbf{x})$ is proportional to the intractable true posterior $p(\mathbf{z} | \mathbf{x})$. To see this, recognise that the optimal proposal is proportional to the nominal distribution times the function whose expectation we are trying to estimate (Owen, 2013). In this case, this specifies that

$$q_\phi^*(\mathbf{z} | \mathbf{x}) \propto p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z}). \quad (2.4)$$

Inspecting and comparing eq. (2.4) to the intractable true posterior inference in eq. (2.1), we can conclude that $q_\phi^*(\mathbf{z} | \mathbf{x}) \propto p_\theta(\mathbf{z} | \mathbf{x})$, which means the optimal proposal is intractable.

2.1.2 Variational inference

To mitigate this, we ditch the optimal posterior proposal and try to learn an approximate posterior $q_\phi(\mathbf{z} | \mathbf{x})$ — a probability distribution parametrised by a neural network with parameters $\phi \in \Phi$. The optimisation goal of our generative model is now two-fold: 1) Maximise the probability of observation $p(\mathbf{x})$ by optimising θ ; 2) learn approximated variational posterior close to the true posterior by optimising ϕ .

This is also known as **amortised inference**, since we are optimising w.r.t. both the variational parameters ϕ and generative parameters θ . Under this setting, the generative model can be represented by the following graph:

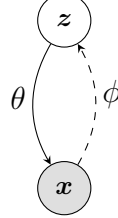


Figure 2.2: Graph representation of VAE. Grey node represents observed variable.

2.1.3 Deriving Evidence Lower Bound (ELBO)

For brevity sake we will omit the parameters that populates the distribution and write $p_\theta(\mathbf{x} | \mathbf{z})$ and $q_\phi(\mathbf{z} | \mathbf{x})$ as $p(\mathbf{x} | \mathbf{z})$, $q(\mathbf{z} | \mathbf{x})$. We can derive the Evidence Lower Bound (ELBO) to the probability $p(\mathbf{x})$ by considering the following: the approximated posterior $q(\mathbf{z} | \mathbf{x})$ should be close to the true posterior $p(\mathbf{z} | \mathbf{x})$. Therefore, the Kullback–Leibler divergence (KL) between the two distributions should be non-negative. This gives us:

$$\begin{aligned}
 & \text{KL}(q(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z} | \mathbf{x})) \\
 &= \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x})}{p(\mathbf{z} | \mathbf{x})} d\mathbf{z} \\
 &= \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x}) p(\mathbf{x})}{p(\mathbf{z} | \mathbf{x}) p(\mathbf{x})} d\mathbf{z} \\
 &= \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x}) p(\mathbf{x})}{p(\mathbf{x}, \mathbf{z})} d\mathbf{z} \\
 &= \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \left(\log q(\mathbf{z} | \mathbf{x}) + \log p(\mathbf{x}) - \log p(\mathbf{x}, \mathbf{z}) \right) d\mathbf{z} \\
 &= \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log p(\mathbf{x}) d\mathbf{z} + \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x})}{p(\mathbf{x}, \mathbf{z})} d\mathbf{z} \\
 &= \log p(\mathbf{x}) - \int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} | \mathbf{x})} d\mathbf{z}. \tag{2.5}
 \end{aligned}$$

Therefore, we have

$$\underbrace{\log p(\mathbf{x})}_{\textcircled{1}} = \underbrace{\text{KL}(q(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z} | \mathbf{x}))}_{\textcircled{2}} + \underbrace{\int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} | \mathbf{x})} d\mathbf{z}}_{\textcircled{3}}. \tag{2.6}$$

Since ② is non-negative, ③ is guaranteed to be smaller than ①. We can rewrite ③ as an expectation over $q(\mathbf{z} \mid \mathbf{x})$, thus giving us

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{z} \mid \mathbf{x})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} \right] \leq \log p(\mathbf{x}). \quad (2.7)$$

By maximising the ELBO, we either 1) maximise the log evidence $\log p(\mathbf{x})$ or 2) improve the variational posterior approximation $q(\mathbf{z} \mid \mathbf{x})$.

Another way of seeing that is a lower bound to the marginal likelihood is by Jensen's inequality. Recall that from eq. (2.3) we have

$$p(\mathbf{x}) = \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} d\mathbf{z}.$$

Recall that this estimation of $p(\mathbf{x})$ gives us an optimal posterior proportional to the intractable true posterior, with function $f(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{z})$ and importance weights $w(\mathbf{x}) = \frac{p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}$. However, by Jensen's inequality we can find the lower bound to the logarithm of eq. (2.3)

$$\log p(\mathbf{x}) = \log \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}) \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} d\mathbf{z} \geq \int_{\mathbf{z}} q(\mathbf{z} \mid \mathbf{x}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} d\mathbf{z} = \mathcal{L}_{\text{ELBO}}, \quad (2.8)$$

which, again, gives us the same ELBO as with section 2.1.3. It is worth noting that from an importance sampling perspective, $\mathcal{L}_{\text{ELBO}}$ has the form of the log of importance-sampled expectation of $f(\mathbf{x}) = 1$, with importance weights $w(\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}$.

2.1.4 Deriving Importance Weighted Autoencoder (IWAE)

Burda et al. (2015) proposes IWAE, which maximises log evidence $\log p(\mathbf{x})$ via the following K -sample ELBO:

$$\mathcal{L}_{\text{IWAE}} = E_{\mathbf{z}^{(1:K)} \sim q(\mathbf{z} \mid \mathbf{x})} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{q(\mathbf{z}^{(k)} \mid \mathbf{x})} \right]. \quad (2.9)$$

As a comparison, we can also write ELBO in section 2.1.3 in its minibatch form, as an average over K samples,

$$\mathcal{L}_{k\text{-ELBO}} = E_{\mathbf{z}^{(1:K)} \sim q(\mathbf{z} \mid \mathbf{x})} \left[\frac{1}{K} \sum_{k=1}^K \log \frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{q(\mathbf{z}^{(k)} \mid \mathbf{x})} \right]. \quad (2.10)$$

Again, by Jensen's inequality, we can easily see $\log p(\mathbf{x}) \geq \mathcal{L}_{\text{IWAE}} \geq \mathcal{L}_{k\text{-ELBO}}$. Apart from posing a tighter lower bound, however, the most prominent benefit of IWAE is that it weighs the samples based on their relative importance. To see this, compare the gradients of the k-sample ELBO and IWAE

$$\nabla \mathcal{L}_{k\text{-ELBO}} = E_{\mathbf{z}^{(1:K)} \sim q(\mathbf{z}|\mathbf{x})} \left[\sum_{k=1}^K \frac{1}{K} \nabla \log \frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{q(\mathbf{z}^{(k)} | \mathbf{x})} \right], \quad (2.11)$$

$$\nabla \mathcal{L}_{\text{IWAE}} = E_{\mathbf{z}^{(1:K)} \sim q(\mathbf{z}|\mathbf{x})} \left[\sum_{k=1}^K \tilde{w}_k \nabla \log \frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{q(\mathbf{z}^{(k)} | \mathbf{x})} \right], \quad (2.12)$$

where

$$\tilde{w}_k = \frac{\frac{p(\mathbf{x}, \mathbf{z}_k)}{q(\mathbf{z}_k | \mathbf{x})}}{\sum_{i=1}^K \frac{p(\mathbf{x}, \mathbf{z}_i)}{q(\mathbf{z}_i | \mathbf{x})}}. \quad (2.13)$$

From eq. (2.11) and eq. (2.12), we see that the ELBO samples are equally weighted during update, while the IWAE samples are weighted by $\tilde{w}_k$.

Interpreting IWAE: the implicit posterior Interestingly, Cremer et al. (2017) show that using IWAE as a lower bound is equivalent to taking samples from an importance-sampling induced implicit proposal distribution q_{IW} . Given K samples from $q(\mathbf{z} | \mathbf{x})$, the importance weighted q_{IW} can be written as a function of one of the samples $\mathbf{z}^{(k)}$ in the following way:

$$\begin{aligned} q_{IW}(\mathbf{z}^{(k)} | \mathbf{x}, \mathbf{z}^{(\setminus k)}) &= k \tilde{w}^{(k)} q(\mathbf{z}^{(k)} | \mathbf{x}) = \left(\frac{\frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{q(\mathbf{z}^{(k)} | \mathbf{x})}}{\frac{1}{K} \sum_{i=1}^K \frac{p(\mathbf{x}, \mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)} | \mathbf{x})}} \right) q(\mathbf{z}^{(k)} | \mathbf{x}) \\ &= \frac{p(\mathbf{x}, \mathbf{z}^{(k)})}{\frac{1}{K} \sum_{i=1}^K \frac{p(\mathbf{x}, \mathbf{z}^{(i)})}{q(\mathbf{z}^{(i)} | \mathbf{x})}} \end{aligned} \quad (2.14)$$

We can see that when $K = 1$, $q_{IW} = q(\mathbf{z} | \mathbf{x})$. When $K > 1$, q_{IW} starts to depend on the true posterior $p(\mathbf{z} | \mathbf{x})$. Cremer et al. (2017) also show that when $K = \infty$, q_{IW} becomes the true posterior $p(\mathbf{z} | \mathbf{x})$ and the inference model is no longer needed. It can also be easily shown that substituting eq. (2.14) for the $q(\mathbf{z} | \mathbf{x})$ in the original ELBO will recover the IWAE bound.

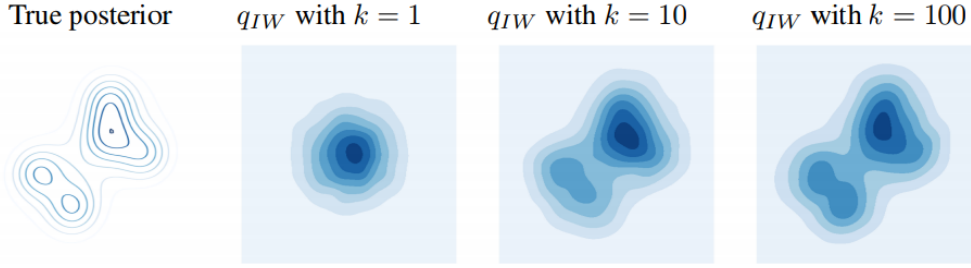


Figure 2.3: Approximation to true posterior with implicit posterior q_{IW} with growing number of K . As K increases, the approximation approaches the true distribution. Image from Cremer et al. (2017).

2.1.5 Gradient variance and its reduction

The analysis above shows that IWAE with $K > 1$ provides a tighter estimator to the true value of $\log p(\mathbf{x})$ than ELBO. Additionally, as we increase the number of particles used K , the variance of the gradient estimator shrinks. However, Rainforth et al. (2018) show that this tighter variational bound, while great for optimising the generative model p_θ , is problematic for the gradient estimation of inference model q_ϕ . To understand this, it is helpful to introduce some of the gradient estimation techniques in variational inference.

Background: gradient estimators

As summarised by Mohamed et al. (2019), gradient estimation of general form $\nabla_\theta \mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})]$ can be computed in two ways:

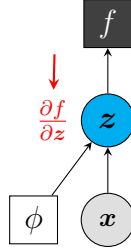
1) Derivative of measure The gradient can be computed by differentiating $p(\mathbf{x}; \theta)$. A well-known approach in this category is the **score function estimator**, which utilises the following "trick" of the rule for log derivative:

$$\nabla_\theta \log p(\mathbf{x}; \theta) = \frac{\nabla_\theta p(\mathbf{x}; \theta)}{p(\mathbf{x}; \theta)} \quad (2.15)$$

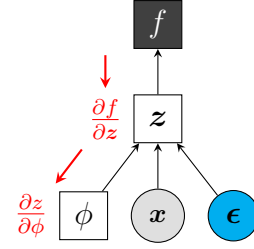
This identity, also known as the REINFORCE trick, is useful as it allows us to write integral of gradients $\nabla_\theta p(\mathbf{x}; \theta)$ as an expectation under the measure p ; it also does not impose any requirement on the differentiability of $f(\mathbf{x})$.

However, it is well known that a number of factors can cause large variance in the score function estimator, including the input dimensionality, choice of cost function

$f(\mathbf{x})$ and importance ratio. Therefore, when we do have access to the derivative of cost $f(x)$, a widely-used gradient estimator is the “reparametrisation trick” (Kingma and Welling, 2014b), which we introduce next –



Original form



Reparametrised form

Figure 2.4: Original form

Figure 2.5: Illustration of how the “reparametrisation trick” allows for gradients flowing through the sampling process at $\mathbf{z}$.

2) Derivative of paths The gradient can be computed by differentiation of $f(\mathbf{x})$. Kingma and Welling (2014b) introduced this principle as the **"reparametrisation trick"** in their first introduction to VAE. The method builds on the principle that continuous distribution allows for both a direct and an indirect way of sampling, making the following processes equivalent:

$$\hat{\mathbf{x}} \sim p(\mathbf{x}; \theta) \quad \equiv \quad \hat{\mathbf{x}} = g(\hat{\epsilon}, \theta), \quad \hat{\epsilon} \sim p(\epsilon). \quad (2.16)$$

By *Law of the Unconscious Statistician* (Grimmett et al., 2001),

$$\mathbb{E}_{p(\mathbf{x}; \theta)}[f(\mathbf{x})] = \mathbb{E}_{p(\epsilon)}[f(g(\epsilon; \theta))] \quad (2.17)$$

This states that the expectation of a function $f(\mathbf{x})$ over a distribution $p(\mathbf{x}; \theta)$ can be computed by sampling from a simpler distribution $p(\epsilon)$, as long as there exists a deterministic mapping g between $p(\epsilon)$ and $p(\mathbf{x}; \theta)$. In the case of VAE, this allows us to sample from the posterior $\mathbf{z} \sim q_\phi(\mathbf{z} | \mathbf{x})$ using $\mathbf{z} = g(\epsilon; \boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi) = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \odot \epsilon$,

where $\epsilon \sim \mathcal{N}(0, 1)$. By doing so, we move the stochasticity of the model from latent z to ϵ , which allows for gradient to flow through ϕ . This is visualised in Figure 2.5.

In summary, path derivative has a much lower variance for gradient estimation compared to the score function. However, the score function has more general applicability as it can be used whenever the loss is not differentiable (e.g. reinforcement learning). In VAEs, the latent distribution is typically specified by mean and variance, so the reparametrisation trick is used to reduce the variance of gradient estimation. However, Roeder et al. (2017) discovers that a hidden score function term in the gradient of ELBO, causing the estimator to have large variance.

To see this, we can first rewrite ELBO as the following:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z) + \log p(z) - \log q_\phi(z|x)] \quad (2.18)$$

We can then take the total derivative of the term within expectation w.r.t. ϕ :

$$\nabla_\phi(\hat{\epsilon}, \phi) = \nabla_\phi [\log p_\theta(x|z) + \log p(z) - \log q_\phi(z|x)] \quad (2.19)$$

$$= \nabla_\phi [\log p_\theta(z|x) + \log p(x) - \log q_\phi(z|x)] \quad (2.20)$$

$$= \underbrace{\nabla_z [\log p_\theta(z|x) - \log q_\phi(z|x)] \nabla_\phi t(\hat{\epsilon}, \phi)}_{\text{path derivative}} - \underbrace{\nabla_\phi \log q_\phi(z|x)}_{\text{score function}} \quad (2.21)$$

Equation (2.21) shows that the gradient w.r.t. ϕ can be expressed as the sum of two components — 1) a **path derivative** component that measures the dependence on ϕ only through the sample z and 2) a **score function** component that measures the dependence on $\log q_\phi$ directly, without considering how the sample z changes as a function of ϕ , causing high variance in the gradient estimator. Rainforth et al. (2018) points out that this problem becomes exacerbated when using IWAE with a large number of posterior samples K .

Several techniques have been proposed to reduce the variance. One such technique is proposed by Roeder et al. (2017), who suggests simply dropping the score function term in the gradient, achieving initial promising results. Another technique is

proposed by Tucker et al. (2019), which involves performing a reparametrisation trick on the score function term, effectively reducing the variance.

2.2 Multimodal Deep Generative Models

This section serves as an additional literature review for multimodal deep generative models, which is discussed in Chapters 3 and 4. Specifically, we organise and summarise methodologies in multimodal VAEs, and perform a literature review on the recent advances on large scale multimodal generative models.

2.2.1 Multimodal VAEs

In Figure 2.6, we show the general schematic for a multimodal VAE with two observed modalities, denoted as $\mathbf{x}$ and $\mathbf{y}$. Data from each modality is fed through its own encoder ϕ_x and ϕ_y , the outputs of which form the joint posterior $q_{\Phi}(z|x, y)$. We then sample from the joint posterior $z \sim q_{\Phi}(z|x, y)$ and pass them through two decoders θ_x, θ_y to reconstruct the two input samples.

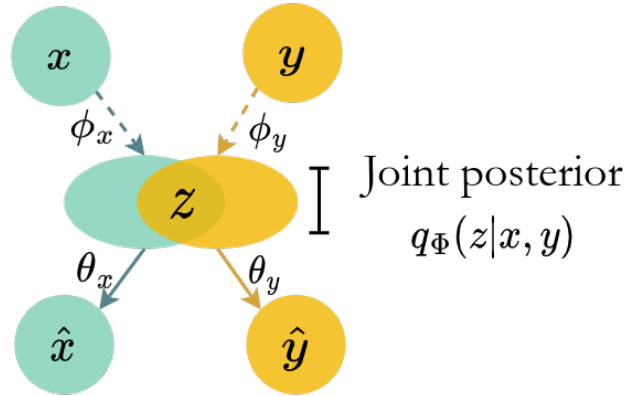


Figure 2.6: A schematic for multimodal VAEs.

Previous approaches on multimodal VAEs mainly differ in how the joint posterior is formed, i.e. how information from different modalities are combined. We divide different approaches into two categories: concatenation and factorisation.

Concatenation Models in this category learn joint representation by either concatenating the inputs themselves or their modality-specific representations. Examples for the former includes early work in multimodal VAEs such as the

JMVAE (Suzuki et al., 2017), triple ELBO (Vedantam et al., 2018) and MFM (Tsai et al., 2019), which define a joint encoder over concatenated multimodal data. Such approaches usually require the training of auxiliary modality-specific components to handle the partially-observed setting, with missing modalities, at test time. They also cannot learn from partially-observed data. In recent work, Gong et al. (2021) propose VSAE where the latent representation is constructed as the concatenation of modality-specific encoders. Inspired by VAEs that deal with imputing pixels in images such as VAEAC (Ivanov et al., 2019), Partial VAE (Ma et al., 2018), MIWAE (Mattei and Frellsen, 2019), HI-VAE (Nazábal et al., 2020) and pattern-set mixture model (Ghalebikesabi et al., 2021), VSAE can learn in the partially-observed setting by incorporating a modality mask. This, however, introduces additional components such as a collective proposal network and a mask generative network, while ignoring the need for the joint distribution over data to capture some notion of the relatedness between modalities.

Factorisation In order to handle missing data at test time without auxiliary components, recent work propose to factorise the posterior over all modalities as the product (Wu and Goodman, 2018b) or mixture (Shi et al., 2019c) of modality-specific posteriors (experts). Following this line of work, Sutter et al. (2021) proposes to combine the two approaches (MoPoE-VAE) to improve learning in settings where the number of modalities exceeds two. In contrast to these methods, mmJSD (Sutter et al., 2020) combines information not in the posterior, but in a “dynamic prior”, defined as a function (either mixture or product) over the modality-specific posteriors as well as pre-defined prior.

Table 2.1 provides a high-level summary of prior work in multimodal VAEs.

2.2.2 Large-scale multimodal generative models

The investigation of learning from multimodal data in larger scale computer vision was partly limited by computational capacity and a lack of understanding in applying deep learning to modalities other than images, such as language, audio, and video. Pioneering this line of research are works in audio-visual learning,

Table 2.1: We examine four characteristics: The ability to handle partial observation at test and train time, the form of the joint distribution or representation in the bi-modal case ($\mathbf{x}$, $\mathbf{y}$ are modalities), and additional components. (✓) indicates a theoretical capability that is not verified empirically.

	Partial Test	Partial Train	Joint repr./dist.	Additional
JMVAE	✓	✗	$q_{\Phi}(\mathbf{z} \mathbf{x}, \mathbf{y})$	$q_{\phi_s}(\mathbf{z} \mathbf{x}), q_{\phi_t}(\mathbf{z} \mathbf{y})$
tELBO	✓	✗	$q_{\Phi}(\mathbf{z} \mathbf{x}, \mathbf{y})$	$q_{\phi_s}(\mathbf{z} \mathbf{x}), q_{\phi_t}(\mathbf{z} \mathbf{y})$
MFV	✓	✗	$q_{\Phi}(\mathbf{z} \mathbf{x}, \mathbf{y})$	$q_{\phi_s}(\mathbf{z} \mathbf{x}), q_{\phi_t}(\mathbf{z} \mathbf{y})$
VSVAE	✓	✓	$\text{concat}(z_s, z_t)$	mask generative network
MVAE	✓	(✓)	$q_{\phi_s}(\mathbf{z} \mathbf{x})q_{\phi_t}(\mathbf{z} \mathbf{y})p(\mathbf{z})$	sub-sampling
MMVAE	✓	✗	$q_{\phi_s}(\mathbf{z} \mathbf{x}) + q_{\phi_t}(\mathbf{z} \mathbf{y})$	-
MoPoE	✓	(✓)	$q_{\phi_s}(\mathbf{z} \mathbf{x}) + q_{\phi_t}(\mathbf{z} \mathbf{y}) + q_{\phi_s}(\mathbf{z} \mathbf{x})q_{\phi_t}(\mathbf{z} \mathbf{y})$	-
mmJSD	✓	✗	$f(q_{\phi_s}(\mathbf{z} \mathbf{x}), q_{\phi_t}(\mathbf{z} \mathbf{y}), p(\mathbf{z}))$	-

featuring video/audio understanding tasks such as audio-visual source separation (Chen et al., 2017; Ghahramani and Jordan, 1995; Hershey et al., 2016; Roweis, 2000; Virtanen, 2007; Yu et al., 2017), audio-visual alignment (Chung and Zisserman, 2017; Marcheret et al., 2015; McAllister et al., 1997; Son Chung et al., 2017), sound localization (Barzelay and Schechner, 2007; Cherry, 1953; Fisher III et al., 2000; Hershey and Movellan, 1999) etc. Another line of work explore a more generative path, such as work on audio synthesis with visual input (Gan et al., 2020; Gao and Grauman, 2019; Koepke et al., 2020) or sound-guided video generation (Chatterjee and Cherian, 2020; Jeong et al., 2021; Le Moing et al., 2021; Yan et al., 2021).

In addition, recent advances in natural language processing have led to a surge in research on large scale visual-language generative models like SimVLM (Wang et al., 2021b), Florence (Yuan et al., 2021), Flamingo (Alayrac et al., 2022), GIT (Wang et al., 2022). These models are primarily designed for tasks like image captioning and visual question answering, but there has also been a growing focus on representation learning for multimodal data in models like CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021b), BEiT (Bao et al., 2021), PaLI (Chen et al., 2022) and CoCa (Yu et al., 2022). These models are found to be more generalisable to unseen distributions, reducing the need for task-specific training data and making them more trustworthy at deployment (Agarwal et al., 2021).

One key feature of these models is that they are typically powered by large vision

and language architectures. For example, the most recent vision-language model PaLI (Chen et al., 2022) from Google contains 17 billion parameters, employing a bespoke Vision Transformer (ViT) model (Dosovitskiy et al., 2021) named ViT-e (“enormous”). In contrast, the architectures used in this thesis are much smaller and the problems studied are less complex. This is partly due to the limited computational resources available at the university, but this constraint also allows for more controlled and well-defined settings, enabling thorough analysis of various factors and the ability to draw convincing conclusions from experiments.

2.3 Domain Generalisation

This section serves as backgrounds to Chapter 5. We discuss the importance of studying domain generalisation, as well as the key ideas behind methods in this area. We also provide a literature review summarising recent advances.

2.3.1 Introduction

Machine learning systems often struggle to generalise outside of their training distribution, leading to poor performance in new environments or conditions. This can manifest in many ways, such as medical systems trained on data from one hospital are unable to effectively diagnose patients in other hospitals, or wealth index prediction models failing to generalise to rural areas where less data is available. This inability to generalise to unseen distribution is often caused by over-reliance on the training data, as well as the learning of irrelevant or spurious correlations.

Domain generalisation aims to tackle with this problem. Typically, the set up involves a training (source) dataset consisting of S domains $\mathcal{D}_s = \{\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(S)}\}$, where each domain s is characterised by a dataset $\mathcal{D}^{(s)} := \{(x_i^{(s)}, y_i^{(s)})\}_{i=1}^{n_s}$ containing data drawn i.i.d. from some probability distribution, and a test dataset $\mathcal{D}_t$ consisting of T domains $\mathcal{D}_t = \{\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(T)}\}$, where $\mathcal{D}_t \cap \mathcal{D}_s = \emptyset$. The goal of domain generalisation is to train a model parametrised by θ on $\mathcal{D}_s$ such that the loss on $\mathcal{D}_t$ is minimised, i.e.

$$\arg \min_{\theta} \mathbb{E}_{\mathcal{D} \sim \mathcal{D}_t} \mathbb{E}_{(x,y) \sim \mathcal{D}} [l((x, y); \theta)]. \quad (2.22)$$

Generalising to unseen distribution can be arbitrarily difficult, as the real world is complex and constantly evolving in its complexity. However, previous research (Arjovsky et al., 2019b; Gulrajani and Lopez-Paz, 2020; Ganin et al., 2016c) have established that this can be achieved by harnessing the multi-source domain set up in the domain generalisation problem, by focusing on features that remain *invariant* across multiple domains, while ignoring those that are correlated with label information in a spurious manner. For instance, a model trained to identify cows and camels using photos taken in different climates must be able to recognise the constant biological features of these animals, rather than relying on correlations with the surrounding landscape (Torralba and Efros, 2011b; Stock and Cisse, 2017). By doing this, the model can have a better chance to generalise to environment that are unseen during training.

2.3.2 Literature review

In domain generalisation, the training data is sampled from one or many source domains, while the test data is sampled from a new target domain. In contrast to domain *adaptation*, the learner does not have access to any data from the target domain (labeled or unlabeled) during train time (Quionero-Candela et al., 2009). We will now discuss the three main families of approaches to domain generalisation:

Distributional Robustness (DRO) DRO approaches minimise the worst-case loss over a set of data distributions constructed from the training domains. Rojas-Carulla et al. (2015) proposed DRO to address *covariate shift* (Gretton et al., 2009a,b), where $P(Y|X)$ remains constant across domains but $P(X)$ changes. Later work also studied *subpopulation shift*, where the train and test distributions are mixtures of the same domains, but the mixture weights change between train and test (Hu et al., 2018; Sagawa et al., 2019);

Domain-invariant representation learning This family of approaches to domain generalisation aims at learning high-level features that make domains statistically indistinguishable. Prediction is then based on these features only. The

principle is motivated by a generalisation error bound for unsupervised domain adaptation (Ben-David et al., 2010; Ganin et al., 2016c), but the approach readily applies to domain generalisation (Gulrajani and Lopez-Paz, 2020; Koh et al., 2020). Algorithms include penalising the domain-predictive power of the model (Ganin et al., 2016c; Wang et al., 2019; Huang et al., 2020), matching mean and variance of feature distributions across domains (Sun and Saenko, 2016b), learning useful representations by solving Jigsaw puzzles (Carlucci et al., 2019) or using the maximum mean discrepancy (Gretton et al., 2006) to match the feature distributions (Li et al., 2018b).

Similar to our proposed gradient matching approach in Shi et al. (2021), Koyama and Yamaguchi (2021a) proposes IGA, which also adopts a gradient-alignment approach for domain generalisation. The key difference between IGA and our IDGM objective is that IGA learns invariant features by minimizing the *variance* of inter-domain gradients. Notably, IGA is completely identical to ERM when ERM is the optimal solution on every training domain, since the variances of the gradients will be zero. While they achieve the best performance on the training set, both IGA and ERM could completely fail when generalizing to unseen domains. Our proposed objective in Shi et al. (2021), on the contrary, biases towards non-ERM solutions as long as the gradients are aligned, and is therefore able to avoid this issue. Additionally, in Lopez-Paz and Ranzato (2017) we also see the application of gradient-alignment, however in this case it is applied under the continual learning setting to determine whether a gradient update will increase the loss of the previous tasks.

Invariant Risk Minimisation (IRM) IRM is proposed by Arjovsky et al. (2019b), which learns an intermediate representation such that the optimal classifiers (on top of this representation) of all domains are the same. The motivation is to exploit invariant causal effects between domains while reducing the effect of domain-specific spurious correlations.

Apart from these algorithms that are tailored for domain generalisation, a well-studied baseline in this area is Empirical Risk Minimisation (ERM), which simply minimises the average loss over training domains. Using vanilla ERM is theoretically unfounded (Hashimoto et al., 2018; Blodgett et al., 2016; Tatman, 2017) since ERM is guaranteed to work only when train and test distributions match. Nonetheless, recent benchmarks suggest that ERM obtains strong performance in practice, in many cases surpassing domain generalisation algorithms (Gulrajani and Lopez-Paz, 2020; Koh et al., 2020). Our goal is to fill this gap, using an algorithm significantly simpler than previous approaches.

Another popular paradigm that studies distribution shift is domain adaptation (DA). Similar to domain generalisation (DG), DA probes the generalisation capabilities of models, however there exists a key distinction between the two paradigms: in DG, models are completely agnostic to the target domain, while in DA unlabelled data from the target domain is given. It is therefore possible in DA for models to leverage the target data distribution during training to improve its generalisation performance at test time. Popular techniques in DA include adversarial training (Ganin et al., 2016a; Tzeng et al., 2017; Bousmalis et al., 2016b), e.g. learning a discriminator such that the representations from source and target domains from the same class are indistinguishable, as well as distance-based methods that aim to align the representations from different domains (Gulrajani and Lopez-Paz, 2020; Li et al., 2020; Balaji et al., 2019; Long et al., 2016a). While DA typically involves only one source domain, more similar to DG, multi-source DA considers multiple source domains (Dong et al., 2021; Nguyen et al., 2021; Park and Lee, 2021; Pei et al., 2018).

2.4 Augmentation-based Self-supervised Learning

In this section we discuss Augmentation-based self-supervised learning (SSL) as context for Chapter 6. Note that while SSL is widely adopted in other perceptual

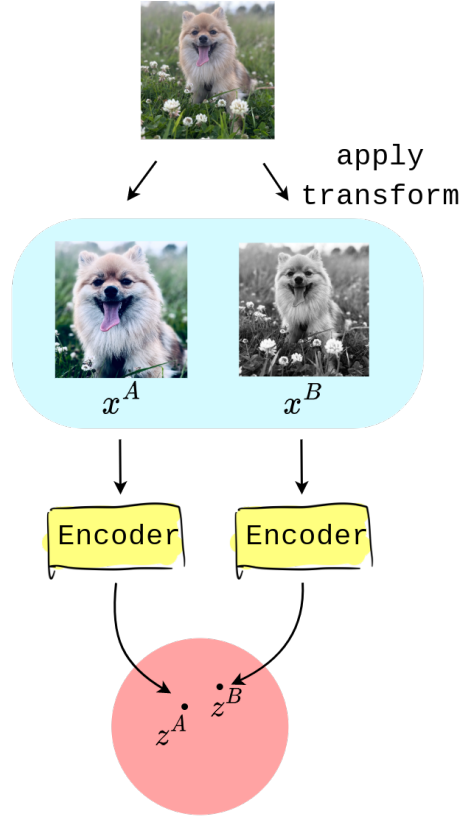


Figure 2.7: A schematic for augmentation-based SSL.

modalities including text, audio and video, here we discuss its application to images only as it is the most relevant to the work presented in Chapter 6.

2.4.1 Introduction

SSL, or self-supervised learning, is a type of machine learning in which the model is trained on a set of pre-defined tasks without the use of explicit labels. The goal of SSL is to learn useful representations from images without the need for labeled data. This is in contrast to supervised learning (SL), in which the model is trained using explicit labels.

SSL methods aim to learn representations of images in an unsupervised manner, without task-specific label information. A way to achieve this is by specifying the following desiderata:

Images that are semantically similar should have representations that are close to each other in feature space.

One cheap way for generating “semantically similar” images is to apply random image augmentations to each data point. Formally, we can define a set of augmentations, represented as T , that includes common techniques such as rotation, cropping, and horizontal flipping. To create two different “views” of an image $\mathbf{x}$, we can use the following process:

$$\begin{array}{ll} t^A \sim T, t^B \sim T & \text{Sample augmentations} \\ \mathbf{x}^A := t^A(\mathbf{x}), \mathbf{x}^B := t^B(\mathbf{x}) & \text{Apply to image} \end{array}$$

As seen in Figure 2.7, the augmentations applied to the image only make stylistic changes and preserves the core information of the image. As a result, the two views, $\mathbf{x}^A$ and $\mathbf{x}^B$, are semantically similar.

To learn the image encoder, we can minimise the representations generated from the two views, $\mathbf{x}^A$ and $\mathbf{x}^B$, using the following objective:

$$z^A := f_\theta(x^A), z^B := f_\theta(x^B) \quad \text{Generate representations} \quad (2.23)$$

$$\min_{\theta} \mathbb{E}[\mathcal{D}(z^A, z^B)] \quad \text{Training objective} \quad (2.24)$$

where $\mathcal{D}(\cdot)$ denotes some distance defined in the vector space of the features $\mathcal{Z}$. The objective in Equation (2.24) allows us to learn the image encoder by following the previously specified criteria for generating representations of semantically similar images that are closed to each other.

While the objective in Equation (2.24) is straightforward, direct minimisation of the objective leads to a phenomenon known as *latent collapse*, where the model maps all images to the same constant representation. To address this issue, Recent works on SSL have proposed effective and innovative solutions, which will be discussed in the following section.

2.4.2 Literature review

SimCLR (Chen et al., 2020b) prevents latent collapse by employing a contrastive objective, where it minimises the distance between representations from two views of the same image, while maximising those between representations from different

images in the same batch. This prevents the model from trivially minimising the objective by projecting all images onto the same point in feature space. MoCo (He et al., 2020b; Chen et al., 2020) also uses a similar contrastive objective, however instead of drawing examples randomly to create negative pairs from the current minibatch, it keeps a dictionary of such negative pairs, thereby decoupling the evaluation of the objective from the minibatch size. In contrast to these methods are the clustering-style self-supervised learning approaches, for instance (Caron et al., 2018) proposes to iteratively cluster representations with k-means and use the cluster assignments as pseudo-labels to train the model; however the iterative update of clusters means that it is difficult to scale this approach to larger datasets where each example is only seen once during training, and the model could suffer from latent collapse. To tackle with these challenges, Caron et al. proposes to enforce consistency of cluster assignments between two different augmented views of the same image.

While the contrastive-style objective proves to be effective, it requires a large number of negative examples to ensure stable training, which can be computationally expensive. Grill et al. (2020) addresses the problem with a method named BYOL, which does not require any negative example to perform updates. The objective of BYOL directly minimise the distance between two representations of two views of the same image, and latent collapse is avoided by not sharing weights between the two encoders of the two views. More specifically, one of the two encoders is updated by gradient descent through minimising the objective, while the other is learned through an online iterative update, effectively the moving average of the other encoder. They show in the paper that by deriving the gradients of this iterative update that latent collapse is avoided as the model is encouraged to utilise stochasticities in training dynamics. Following similar motivations, W-MSE (Ermolov et al., 2021) also removes the need of using negative examples by performing whitening on the representation learned, which maps all representations to a unit sphere to avoid latent collapse; Barlow Twins (Zbontar et al., 2021) on the other hand, avoids latent collapse by matching the cross-correlation matrix between

the representations of two views to an identity matrix, while VICReg (Bardes et al., 2021) does so by regularising the standard deviation of the feature space. SimSiam (Chen and He, 2021) further summarises the advances in prior work and discovers that latent collapse can be effectively prevented without negative examples by utilising momentum encoders while using a large mini-batch size. Following this body of work that utilises knowledge distillation to avoid latent collapse, DINO (Caron et al., 2021) also proposes to learn one of the encoders through iterative update as exponential moving average of the other encoder; they also adopt vision transformer (Dosovitskiy et al., 2021) (instead of ConvNets as was standard at the time) and demonstrate positive results.

3

Variational Mixture-of-Experts Autoencoders for Multimodal Deep Generative Models

Abstract

Learning generative models that span multiple data modalities, such as vision and language, is often motivated by the desire to learn more useful, generalisable representations that faithfully capture common underlying factors between the modalities. In this work, we characterise successful learning of such models as the fulfilment of four criteria: i) implicit latent decomposition into shared and private subspaces, ii) coherent joint generation over all modalities, iii) coherent cross-generation across individual modalities, and iv) improved model learning for individual modalities through multimodal integration. Here, we propose a mixture-of-experts multimodal variational autoencoder (MMVAE) to learn generative models on different sets of modalities, including a challenging image $\leftrightarrow$ language dataset, and demonstrate its ability to satisfy all four criteria, both qualitatively and quantitatively. Code, data, and models are provided at this url.

3.1 Introduction

Human learning in the real world involves a multitude of perspectives of the same underlying phenomena, such as perception of the same environment through visual observation, linguistic description, or physical interaction. Given the lack of explicit labels available for observations in the real world, observing *across modalities* can provide

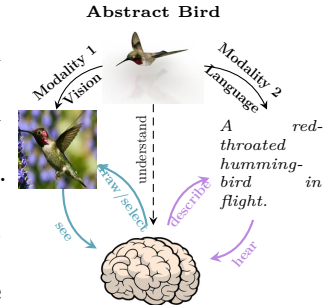


Figure 3.1: A schematic for multimodal perception.

important information in the form of correlations between the observations. Studies have provided evidence that the brain jointly embeds information across different modalities (Stein et al., 2009; Quiroga et al., 2009), and that such integration benefits reasoning and understanding through expression along these modalities (Bauer and Johnson-Laird, 1993; Fan et al., 2018), further facilitating information transfer between (Yildirim, 2014) them. We take inspiration from this to design algorithms that handle such multimodal observations, while being capable of a similar breadth of behaviour. Figure 3.1 shows an example of such a situation, where an abstract notion of a bird is perceived through both visual observation as well as linguistic description. A thorough understanding of what a bird is involves understanding not just the characteristics of its visual and linguistic features individually, but also how they relate to each other (Barsalou, 2008; Siskind, 1994). Moreover, demonstrating such understanding involves being able to visualise, or discriminate birds against other things, or describe birds’ attributes. Crucially, this process involves flow of information in *both* ways—from observations to representations and vice versa.

With this in mind, when designing algorithms that imitate the human learning process, we seek a generative model that is able to jointly embed and generate multimodal observations, which learn concepts by association of multiple modalities and feedback from their reconstructions. While the Variational Autoencoder (VAE) (Kingma and Welling, 2014b) fits such a description well, truly capturing the range of behaviour and abilities exhibited by humans from multimodal observation

requires enforcing particular characteristics on the framework itself. Although there have been a range of approaches that broadly tackle the issue of multimodal generative modelling (c.f. section 3.2), they fall short of expressing a more complete range of expected behaviour in this setting. We hence posit four criteria that a multimodal generative model should satisfy (c.f. fig. 3.2[left]):

Latent Factorisation: the latent space implicitly factors into subspaces that capture the *shared* and *private* aspects of the given modalities. This aspect is important from the perspective of downstream tasks, where better decomposed representations (Lipton, 2018; Mathieu et al., 2019a) are more amenable for use on a wider variety of tasks.

Coherent Joint Generation: generations in different modalities stemming from the same latent value exhibit coherence in terms of the shared aspects of the latent. For example, in the schematic in fig. 3.1, this could manifest through the generated image and description always matching semantically—that is, the description is true of the image.

Coherent Cross Generation: the model can generate data in one modality conditioned on data observed in a different modality, such that the underlying commonality between them is preserved. Again taking fig. 3.1 as an example, for a given description, one should be able to generate images that are semantically consistent with the description, and vice versa.

Synergy: the quality of the generative model for any single modality is improved through representations learnt across multimodal observations, as opposed to just the single modality itself. That is, observing both the image and description should lead to more specificity in generation of the images (and descriptions) than when taken alone.

To this end, we propose the MMVAE, a multimodal VAE that uses a Mixture-of-Expert (MOE) variational posterior over the individual modalities to learn a multimodal generative model that satisfies the above four criteria. While it shares some characteristics with the most recent work on multimodal generative models

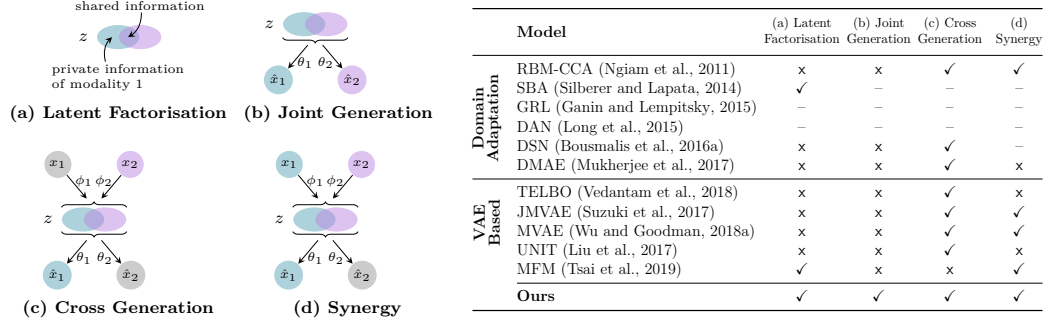


Figure 3.2: [Left] The four criteria for multimodal generative models: (a) latent factorisation (b) coherent joint generation (c) coherent cross generation and (d) synergy. [Right] A characterisation of recent work that explores multiple modalities, including our own, in terms of the specified criteria. See section 3.2 for further details.

(cf. section 3.2), it nonetheless differs from them in two important ways. First, we are interested in situations where 1) observations across multiple modalities are always presented during training; 2) trained model can handle missing modalities at test time. Second, our experiments use many-to-many multimodal mapping scenario, which provides a greater challenge than the typically used one-to-one image $\leftrightarrow$ image (colourisation or edge/outline detection) and image $\leftrightarrow$ attribute transformations (image classification). To the best of our knowledge, we are also the first to explore image $\leftrightarrow$ language transformation under the multimodal VAE setting. Figure 3.2[right] summarises relevant work (cf. section 3.2) and identifies if they satisfy our proposed criteria.

3.2 Related Work

Cross-modal generation Prior approaches to generative modelling with multimodal data have predominantly only targetted cross-modal generation. Given data from two domains $\mathbf{x}_1$ and $\mathbf{x}_2$, they learn the conditional generative model $p(\mathbf{x}_1 | \mathbf{x}_2)$, where the conditioning modality $\mathbf{x}_2$ and generation modality $\mathbf{x}_1$ are typically *not* interchangeable. This is commonly seen in conditional VAE methods for attribute $\rightarrow$ image or image $\rightarrow$ caption generation (Pandey and Dukkipati, 2017; Yan et al., 2016; Sohn et al., 2015; Pu et al., 2016), as well as Generative Adversarial Network (GAN)-based models for cross domain image-to-image translation (Ledig et al., 2017; Li and Wand, 2016; Wang and Gupta, 2016; Taigman et al., 2017; Liu

et al., 2019). In recent years, there have been a few approaches involving both VAEs and GANs that enable cross-modal generation *both* ways (Wang et al., 2016; Zhu et al., 2017a,b), but ignore learning a common embedding between them, instead treating the different cross-generations as independent but composable transforms. Curiously some GAN cross-generation models appear to avoid learning abstractions of data, choosing instead to hide the actual input directly in the high-frequency components of the output (Chu et al., 2017).

Domain adaptation The related sub-field of domain adaptation explores learning joint embeddings of multimodal observations that generalise across the different modalities for both classification (Long et al., 2015; Tzeng et al., 2014; Long et al., 2016b) and generation (Bousmalis et al., 2016a) tasks. And regarding approaches that go beyond just cross-modal generation to models that learn common embedding or projection spaces, Ngiam et al. (2011) were the first to employ an autoencoder-based architecture to learn joint representation between modalities, using their RBM-CCA model. Silberer and Lapata (2014) followed this with a stacked autoencoder architecture to jointly embed the visual and textual representations of nouns. Tian and Engel (2019) considers an intermediary embedding space to transform between *independent* VAEs, applying constraints on the closeness of embeddings and their individual classification performance on labels.

Joint models Yet another class of approaches attempt to explicitly model the joint distribution over latents and data. Suzuki et al. (2017) introduced the joint multimodal VAE (JMVAE) that learns shared representation with joint encoder $q_\phi(\mathbf{z} \mid \mathbf{x}_1, \mathbf{x}_2)$. To handle missing data at test time, two unimodal encoders $q_\phi(\mathbf{z} \mid \mathbf{x}_1)$ and $q_\phi(\mathbf{z} \mid \mathbf{x}_2)$ are trained to match $q_\phi(\mathbf{z} \mid \mathbf{x}_1, \mathbf{x}_2)$ with a KL constraint between them. Vedantam et al. (2018)’s multimodal VAE TELBO (triple ELBO) also deals with missing data at test time by explicitly defining multimodal and unimodal inference networks. However, differing from the JMVAE, they facilitate the convergence between the unimodal encoding distributions and the joint distribution using a two-step training regime that first fits the joint encoding, and subsequently fits the unimodal encodings holding the joint fixed. Tsai et al. (2019) propose

MFM (multimodal factorisation model), which also explicitly defines a joint network $q_\phi(\mathbf{z} \mid \mathbf{x}_{1:M})$ on top of the unimodal encoders, seeking to infer missing modalities using the observed modalities.

We argue that these approaches are less than ideal, as they typically only target *one* of the proposed criteria (e.g. (one-way) cross-generation), often require additional modelling components and inference steps (JMVAE, TELBO, MFM), ignore the latent representation structures induced and largely only target observations within a particular domain, typically vision $\leftrightarrow$ vision (Liu et al., 2017). More recently, Wu and Goodman (2018a) introduced the MVAE, a marked improvement over previous approaches, proposing to model the joint posterior as a Product-of-Expert (PoE) over the marginal posteriors, enabling cross-modal generation at test time *without* requiring additional inference networks and multi-stage training regimes. While already a significant step forward, we observe that the PoE factorisation does not appear to be practically suited for multimodal learning, likely due to the precision miscalibration of experts. See Section 3.3 for more detailed explanation. We also observe that latent-variable mixture models have previously been applied to generative models targetting multimodal topic-modelling (Barnard et al., 2003; Blei and Jordan, 2003). Although differing from our formulation in many ways, these approaches nonetheless indicate the suitability of mixture models for learning from multimodal data.

3.3 Methods

Background We employ a VAE (Kingma and Welling, 2014b) to learn a multimodal generative model over modalities $m = 1, \dots, M$ of the form $p_\theta(\mathbf{z}, \mathbf{x}_{1:M}) = p(\mathbf{z}) \prod_{m=1}^M p_{\theta_m}(\mathbf{x}_m \mid \mathbf{z})$, with the likelihoods $p_{\theta_m}(\mathbf{x}_m \mid \mathbf{z})$ parametrised by deep neural networks (decoders) with parameters $\Theta = \{\theta_1, \dots, \theta_M\}$. The objective of training VAEs is to maximise the marginal likelihood of the data $p_\theta(\mathbf{x}_{1:M})$. However, computing the evidence is intractable as it requires knowledge of the true joint posterior $p_\theta(\mathbf{z} \mid \mathbf{x}_{1:M})$. To tackle this, we approximate the true *unknown* posterior by a variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x}_{1:M})$, which now allows optimising an Evidence

Lower Bound (ELBO) through Stochastic Gradient Descent (SGD), with ELBO defined as

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}_{1:M}) = \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x}_{1:M})} \left[\log \frac{p_{\theta}(\mathbf{z}, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M})} \right] \quad (3.1)$$

The Importance Weighted Autoencoder (IWAE) (Burda et al., 2015) computes a tighter lower bound through appropriate weighting of a multi-sample estimator, as

$$\mathcal{L}_{\text{IWAE}}(\mathbf{x}_{1:M}) = \mathbb{E}_{\mathbf{z}^{1:K} \sim q_{\phi}(\mathbf{z}|\mathbf{x}_{1:M})} \left[\log \sum_{k=1}^K \frac{1}{K} \frac{p_{\theta}(\mathbf{z}^k, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z}^k | \mathbf{x}_{1:M})} \right] \quad (3.2)$$

Beyond the targetting of a tighter bound, we further prefer the IWAE estimator since the variational posteriors it estimates tend to have higher entropy (see Appendix B.4). This is actually beneficial in the multimodal learning scenario, as each posterior $q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)$ is encouraged to assign high probability to regions *beyond* just those which characterise its own modality m .

The Mixture-of-Expert (MoE) joint variational posteriors Given the objective, a crucial question however remains: how should we learn the variational joint posterior $q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M})$?

One immediately obvious approach is to train one single encoder network that takes all modalities $\mathbf{x}_{1:M}$ as input to explicitly parametrise the joint posterior. However, as described in Section 3.2, this approach requires all modalities to be presented at all times, thus making cross-modal generation difficult. We propose to factorise the joint variational posterior as a combination of unimodal posteriors, using a mixture of experts (MoE), i.e. $q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M}) = \sum_m \alpha_m \cdot q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)$, where $\alpha_m = 1/M$, assuming the different modalities are of comparable complexity (as per motivation in Section 3.1).

MoE vs. PoE An alternative choice of factorising the joint variational posterior is as a product of experts (PoE), i.e. $q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M}) = \prod_m q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)$, as seen in MVAE (Wu and Goodman, 2018a). When employing PoE, each expert holds the power of veto—in the sense that the joint distribution will have low density for a

given set of observations if just one of the marginal posteriors has low density. In the case of Gaussian experts, as is typically assumed¹, experts with greater precision will have more influence over the combined prediction than experts with lower precision. When the precisions are miscalibrated, as likely in learning with SGD due to difference in complexity of input modalities or initialisation conditions, overconfident predictions by one expert—implying a potentially biased mean prediction overall—can be detrimental to the whole model. This can be undesirable for learning factored latent representations across modalities. By contrast, MoE does not suffer from potentially overconfident experts, since it effectively takes a vote amongst the experts, and spreads its density over all the individual experts. This characteristic makes them better-suited to latent factorisation, being sensitive to information across all the individual modalities. Moreover Wu and Goodman (2018a) noted that PoE does not work well when observations across all modalities are always presented during training, requiring artificial subsampling of the observations to ensure that the individual modalities are learnt faithfully. As evidence, we show empirically in section 3.4 that the PoE factorisation does not satisfy all the criteria we outline in section 3.1.

The MoE-multimodal VAE (MMVAE) objective With the MoE joint posterior, we can extend the $\mathcal{L}_{\text{IWAE}}$ in Equation (3.2) to multiple modalities by employing stratified sampling (Robert and Casella, 2013) to average over M modalities:

$$\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}) = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{z}_m^{1:K} \sim q_{\phi_m}(z|\mathbf{x}_m)} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p_{\theta}(\mathbf{z}_m^k, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z}_m^k | \mathbf{x}_{1:M})} \right], \quad (3.3)$$

which has the effect of weighing the gradients of samples from different modalities *equally* while still estimating tight bounds for each individual term. Note that although an even tighter bound can be computed by weighting the contribution of each modality differently, in proportion to its contribution to the marginal likelihood,

¹Training PoE models in general can be intractable (Hinton, 2002a) due to the required normalisation, but becomes analytic when the experts are Gaussian.

doing so can lead to undesirable modality dominance similar to that in the PoE case. See Appendix B.1 for further details and results.

It is easy to show that $\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$ is still a tighter lower bound than the standard M -modality ELBO using linearity of expectations, as

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}_{1:M}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_{1:M})} \left[\log \frac{p_\theta(\mathbf{z}, \mathbf{x}_{1:M})}{q_\phi(\mathbf{z} | \mathbf{x}_{1:M})} \right] = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{z}_m \sim q_{\phi_m}(\mathbf{z}|\mathbf{x}_m)} \left[\log \frac{p_\theta(\mathbf{z}_m, \mathbf{x}_{1:M})}{q_\phi(\mathbf{z}_m | \mathbf{x}_{1:M})} \right] \\ &\leq \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{z}_m^{1:K} \sim q_{\phi_m}(\mathbf{z}|\mathbf{x}_m)} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(\mathbf{z}_m^k, \mathbf{x}_{1:M})}{q_\phi(\mathbf{z}_m^k | \mathbf{x}_{1:M})} \right] = \mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}).\end{aligned}$$

In actually computing the gradients for the objectives in Equations (3.3) and (B.1) we employ the DReG IWAE estimator of Tucker et al. (2019), avoiding issues with the quality of the estimator for large K as discovered by Rainforth et al. (2018) (cf. Appendix B.3).

From a computational perspective, the MoE objectives incur some overhead over the PoE objective, due to the fact that each modality provides samples from its own encoding distribution $q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)$ to be evaluated with the joint generative model $p_\theta(\mathbf{z}, \mathbf{x}_{1:M})$, needing M^2 passes over the respective decoders in total. The real cost of such added complexity however can be minimal since a) the number of modalities one can simultaneously process is typically quite small, and b) the additional computation can be efficiently vectorised. However, if this cost should be deemed prohibitively large, one can in fact trade off the tightness of the estimator for linear time complexity in the number of modalities M , employing a multimodal importance sampling scheme on the standard ELBO. We discuss this in further detail in appendix B.2.

3.4 Experiments

To evaluate our model, we constructed two multimodal scenarios to conduct experiments on. The first experiment involves many-to-many image $\leftrightarrow$ image transforms on matching digits between the MNIST and street-view house numbers (SVHN) datasets. This experiment was designed to separate perceptual complexity (i.e. color, style, size) from conceptual complexity (i.e. digits) using relatively simple

image domains. The second experiment involves a highly challenging image $\leftrightarrow$ language task on the Caltech-UCSD Birds (CUB) dataset—more complicated than the typical image $\leftrightarrow$ attribute transformations employed in prior work. We choose this dataset as it matches our original motivation in tackling multimodal perception in a similar manner to how humans perceive and learn about the world. For each of these experiments, we provide both qualitative and quantitative analyses of the extent to which our model satisfies the four proposed criteria—which, to reiterate, are i) implicit latent decomposition, ii) coherent joint generation over all modalities, iii) coherent cross-generation across individual modalities, and iv) improved model learning for individual modalities through multimodal integration. Source code for all models and experiments is available at <https://github.com/iffsid/mmvae>.

3.4.1 Common Details

Across experiments, we employ Laplace priors and posteriors, constraining their scaling across the D dimensions to sum to D . These design choices better encourage the learning of axis-aligned representations by breaking the rotationally-invariant nature of the standard isotropic Gaussian prior (Mathieu et al., 2019a). For learning, we use the Adam optimiser (Kingma and Ba, 2014) with AMSGrad (Reddi et al., 2018), with a learning rate of 0.001. Details of the architectures used are provided in Appendix B.6. All numerical results were averaged over 5 independently trained models. Data and pre-trained models from our experiments are also available at <https://github.com/iffsid/mmvae>.

3.4.2 MNIST-SVHN

Dataset: As mentioned before, we design this experiment in order to probe *conceptual* complexity separate from *perceptual* complexity. We construct a dataset of pairs of MNIST and SVHN such that each pair depicts



Figure 3.3: Example data the same digit class. Each instance of a digit class (in either dataset) is randomly paired with 20 instances of the same digit class from the other dataset. As shown in fig. 3.3, although the data domains are fairly well known, effectively capturing the

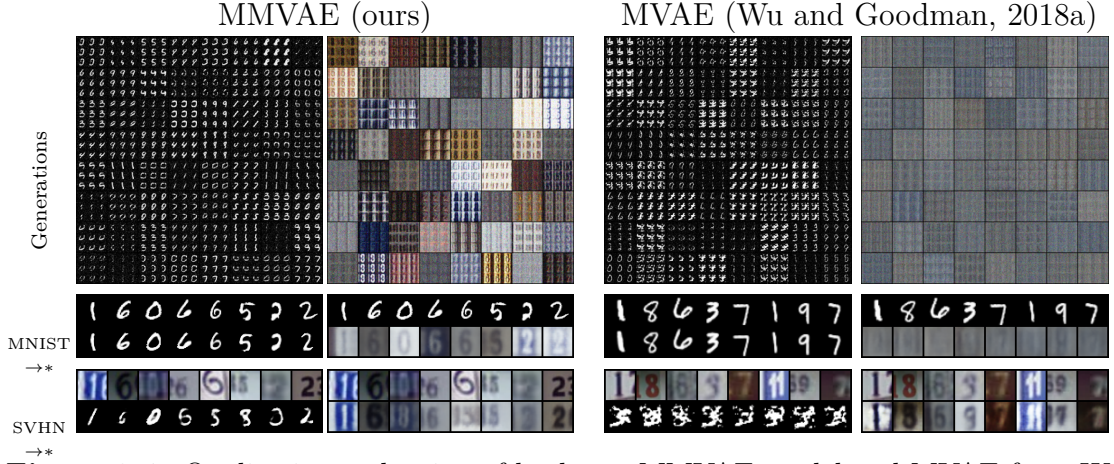


Figure 3.4: Qualitative evaluation of both our MMVAE model and MVAE from Wu and Goodman (2018a). Generations (top row) for each modality. Note both the quality of generations and the extent to which corresponding generations in MNIST and SVHN match on digits for MMVAE vs. MVAE, satisfying the coherent generation criteria. Reconstructions and cross-generations for MNIST (middle row) and SVHN (bottom row). Again note the extent to which cross generations capture the underlying digit effectively for MMVAE.

digit class can be a challenging task due to the variety of styles and colours presented across both datasets. Here, we use CNNs for SVHN and MLPs for MNIST, with a $20d$ latent space.

Qualitative Results: Figure 3.4 shows a qualitative comparison between MMVAE trained with the MOE objective in eq. (3.3) against the MVAE model of Wu and Goodman (2018a). The MVAE was trained using the authors’ publicly available code², following their recommended training regime. We generate from the model $p_\theta(\mathbf{z}, \mathbf{x}_{1:M})$ by taking $R = 64$ samples from the prior $\mathbf{z} \sim p(\mathbf{z})$, each of which is used to take $N = 9$ samples from the likelihood of each modality m as $x_m \sim p_\theta(\mathbf{x}_m | \mathbf{z})$. Note the quality of the MMVAE model, both at coherent joint generation (top row)—where corresponding samples for the same $\mathbf{z}$ match in their digits—and at coherent cross-generation (middle and bottom rows). To show that it is truly the MOE factorisation that impacts the learning—rather than our particular choice of model architecture or the IWAE objective—we explore performance on *only adopting MOE*, directly in the codebase of Wu and Goodman (2018a), keeping all other aspects fixed, in Appendix B.5. Results indicate MVAE with the MOE

²<https://github.com/mhw32/multimodal-vae-public>

posterior does appear to do better, especially at cross-modal generation, than the PoE.

We subsequently analyse the structure of the latent space by traversing each dimension independently as shown in fig. 3.5. Here, for each modality m , we encode datapoint $\mathbf{x}_m$ through its respective encoder $q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)$ to obtain the mean embedding μ_m . Then, we perturb the embedding value along each dimension μ_m^d linearly in the range $(\mu_m^d - 5\sigma_0^d, \mu_m^d + 5\sigma_0^d)$, where σ_0^d is the *learned* standard deviation for dimension d in the prior $p(\mathbf{z})$. Note the extent to which particular dimensions affect only a single modality, whereas other dimensions affect both, indicating a degree of latent factorisation. Also shown is a plot of the per-dimension Kullback–Leibler divergence (KL) between each posterior and the prior, as well as the symmetric KL between the two posteriors, to indicate which dimensions encode information from which posterior, if any.

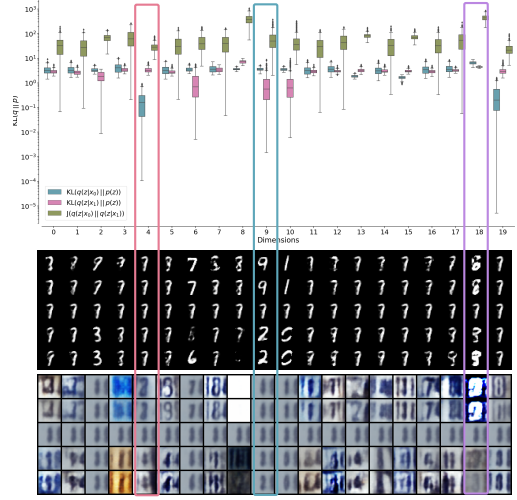


Figure 3.5: Per-dimension latent traversals for a pair of datapoints indicating dimensions that affect only **SVHN**, only **MNIST**, and both **MNIST & SVHN**.

Quantitative Results: To quantify the extent to which the latent spaces factorises from multimodal observations, we employ a simple linear classifier on the latent representations as we have no *a-priori* reason to believe that the representations factorise in an axis-aligned manner. If a linear digit classifier can extract the digit information from the shared latent space, it strongly indicates the presence of a linear subspace that has factored as desired. We train digit classifiers for a) MMVAE, b) MVAE, with posterior either from single-modality inputs or multimodality inputs, and c) single-VAE that takes input from one modality only, plotting results in Table 3.1. Comparing the results of the first column to the last in Table 3.1, we find MMVAE’s latent space provides significantly better accuracy over the single-VAE. For the MVAE, due to its PoE formulation, it appears that the MNIST modality dominates, obtaining high accuracy for MNIST digit classification

(95.7%) but low for SVHN (9.10%). When given *both* inputs, accuracy for SVHN improves significantly while that for MNIST decreases slightly. Note that the accuracy for MVAE is higher when only the MNIST data is presented compared to when both modalities are available, indicating that the presence of the extra modality does not further inform the model on the classification of digits.

Table 3.1: Digit classification accuracy (%) of latent variables in different models.

	MMVAE	MVAE (single)	MVAE (both)	single-VAE
MNIST	91.3	95.7	94.9	85.3
SVHN	68.0	9.1	90.1	20.7

We also quantify the *coherence* of joint generations and cross-modal generations. To do so, we employ off-the-shelf digit classifiers of the original MNIST and SVHN datasets on the generation results, and compute a) for joint generation, how often the digits of generations in two modalities match, and b) for cross-modal generation, how often the digit generated in one modality matches its input from another modality. Results in Table 3.2 indicate that for joint generation, the classifiers predict the same digit class 42.1% of time, and for cross-generation, 86.4% (MNIST $\rightarrow$ SVHN) and 69.1% (SVHN $\rightarrow$ MNIST). Computing these metrics for MVAE yields accuracy close to chance, suggesting that the coherence between modalities is not quite preserved when considering generation.

Table 3.2: Probability of digit matching (%) for joint and cross generation.

	Joint	Cross (M $\rightarrow$ S)	Cross (S $\rightarrow$ M)
MMVAE	42.1	86.4	69.1
MVAE	12.7	9.5	9.3

We finally compute the marginal likelihoods³ of the joint generative model $p_{\theta}(\mathbf{x}_{1:M})$, and each of the individual generative models $p_{\theta_m}(x_m)$ using both the *joint* variational posterior $q_{\phi}(\mathbf{z} \mid \mathbf{x}_{1:M})$ and the *single* variational posterior $q_{\phi_m}(\mathbf{z} \mid x_m)$ as shown in Table 3.3.

³We compute a 1000-sample estimate using eq. (B.1) here.

Table 3.3: Evaluating the different log likelihoods for different arrangements of MNIST and SVHN.

		$\log p(\mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m)$	$\log p(\mathbf{x}_m \mathbf{x}_n)$
$m = \text{MNIST}$	MMVAE	6261.40	868.76	868.37	628.31
$n = \text{SVHN}$	MVAE	2961.80	-176.68	-107.46	-778.20
$m = \text{SVHN}$	MMVAE	6261.40	3441.01	3441.01	2337.56
$n = \text{MNIST}$	MVAE	2961.80	3395.12	3536.86	-12747.50

We observe that MMVAE model yields higher likelihoods, consistent with employing the IWAE estimator. Interestingly, we observe that $p(\mathbf{x}_m | \mathbf{x}_m, \mathbf{x}_n) \geq p(\mathbf{x}_m | \mathbf{x}_m)$ for MMVAE, whereas for MVAE we consistently find $p(\mathbf{x}_m | \mathbf{x}_m, \mathbf{x}_n) < p(\mathbf{x}_m | \mathbf{x}_m)$. This serves to highlight that the MMVAE model is able to effectively utilise information jointly across multiple modalities, which the MVAE model potentially suffers from overdominant encoders and an ill-suited sub-sampled training scheme to accommodate data always present across modalities at train time.

3.4.3 CUB Image-Captions

Dataset: Encouraged by the results from our previous experiment, we consider a multimodal experiment more in line with our original motivation. We employ the images and captions from Caltech-UCSD Birds (CUB) dataset (Wah et al., 2011), containing 11,788 photos of birds in natural scenes, each annotated with

10 fine-grained captions describing the bird’s appearance characteristics collected through Amazon Mechanical Turk (AMT). As shown in fig. 3.6, the images are quite detailed and descriptions fairly complex, involving the composition of various attributes. For the image data, rather than generating directly in image space, we instead generate in the feature space of a pre-trained ResNet-101 (He et al., 2016), in order to avoid issues with blurry generations for complex image data (Zhao et al., 2017). For generations and reconstructions, we simply perform a nearest-neighbour lookup in feature space using Euclidean distance on the generated or reconstructed feature. For the language data, we employ a CNN encoder and decoder following



Figure 3.6: Example data

Kalchbrenner et al. (2014); Pham et al. (2016); Massiceti et al. (2018b), learning an embedding for words in the process. We use 128-dimensional latents with a Laplace likelihood on image features and a Categorical likelihood for captions.



Figure 3.7: Qualitative evaluation of the MMVAE model on the CUB data, showing reconstruction in the individual modalities (top rows), cross generation (middle and bottom rows on right), and joint generation (bottom row on left). More qualitative examples can be found in Appendix B.7.

Qualitative Results: Figure 3.7 shows qualitative results for the MMVAE model trained with the MOE objective in eq. (3.3) on the CUB data. We generate from the model as before, with $R = 4$, and $N_1 = 9, N_2 = 1$. Interestingly, even for such a complicated dataset, we see joint generation align quite well with descriptions largely in line with the image for a range of different attributes, and cross generation where the descriptions again match the image quite well and vice versa. An interesting avenue to explore for future directions in this experiment would be the incorporation of more complex generative models for images potentially incorporating the use of GANs to improve the quality of generated output, and observing its effect on the joint modelling. We also generate results for MVAE on this dataset, where we observe that the image modality dominates and resulting in poor language

reconstruction and cross-modal generation. See Appendix B.8 for examples and further details.

Quantitative Results: We evaluate the coherence of our generation by calculating the correlation between the jointly and cross-generated image-caption pair. We do so by employing Canonical Correlation Analysis (CCA) following the observation of its effectiveness as a baseline for language and vision tasks by Massiceti et al. (2018a). Here, given paired observations $\{x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}\}$, CCA learns projections $W_1 \in \mathbb{R}^{n_1 \times k}$ and $W_2 \in \mathbb{R}^{n_2 \times k}$ that maximise the correlation between projected variables $W_1^T x_1$ and $W_2^T x_2$. With this formulation, the correlation between any new pair of observations $\{\tilde{x}_1, \tilde{x}_2\}$ can be computed as the cosine distance between the mean-centered projected variables, i.e.

$$\text{corr}(\tilde{x}_1, \tilde{x}_2) = \frac{\phi(\tilde{x}_1)^T \phi(\tilde{x}_2)}{\|\phi(\tilde{x}_1)\|_2 \|\phi(\tilde{x}_2)\|_2} \quad (3.4)$$

where $\phi(\tilde{x}_n) = W_n^T \tilde{x}_n - \text{avg}(W_n^T x_n)$.

We prepare the dataset for CCA by pre-processing both modalities using feature extractors. For images, similar to training, we use the off-the-shelf ResNet-101 to generate feature vector of dimension $2048-d$; for captions, we fit a FastText model on all sentences in the training set, projecting each word onto a $300-d$ vector (Bojanowski et al., 2017). The representation for each caption is then acquired by aggregating the embedding of all words in the sentence (here we simply take the average). To compute the correlation between the generated images and captions, we first compute the projection matrix W for each modality using the training set of CUB Image-Captions, then perform CCA using eq. (3.4), on i) jointly generated image-sentence pairs, taking average over 1000 examples, and ii) image-sentence or sentence-image pair of cross generation, taking an average over the entire test set. Results are as shown in Table 3.4.

Table 3.4 shows that the average correlation of joint generation of our model is 0.263; This value is only slightly lower than the average correlation of the data itself (0.273 in Table 3.4), which demonstrates the high coherence between the

Table 3.4: Correlation of Image (I)-Sentence (S) pair for joint and cross generation.

	Joint	Cross (I $\rightarrow$ S)	Cross (S $\rightarrow$ I)	Ground Truth
MMVAE	0.263	0.104	0.135	0.273
MVAE	-0.095	0.011	-0.013	

jointly generated image-caption pairs. For cross-generation, the correlation between *input* images and *generated* captions is slightly lower than that of *input* caption and *generated* image, evaluated at 0.104 and 0.135 respectively.

In comparison, the MVAE model appears to provide (marginally) negative correlation for the jointly generated pairs and sentence $\rightarrow$ image cross generation pairs. Notably, the correlation for image $\rightarrow$ sentence generation is much higher than sentence $\rightarrow$ image (0.011 and -0.013 respectively). Observing the qualitative results for MVAE in Appendix B.8 also shows that any outputs generated from images as input is more expressive than those from the language inputs. These findings indicate that the model places more weight on the image modality than language for the factorisation of joint posterior, once again indicating an overdominant encoder, providing empirical evidence for our analysis of the POE’s potential bias towards stronger experts in Section 3.3.

3.5 Discussions

In this paper, we explore multimodal generative models, characterising successful learning of such models as the fulfillment of four specific criteria: i) implicit latent decomposition into shared and private subspaces, ii) coherent simultaneous joint generation over all modalities, iii) coherent cross-generation between individual modalities, and iv) improved model learning for the individual modalities as a consequence of having observed data from different modalities. Satisfying these goals enables more useful and generalisable representations for downstream tasks such as classification, by capturing the abstract relationship between the modalities. To this end, we propose a variational Mixture-of-Expert (MOE) autoencoder framework that allows us to achieve these criteria, in contrast to prior work which primarily target just the cross-modal generation and improved model learning

aspects. We compare and contrast our MMVAE model against the state-of-the-art Product-of-Expert (PoE) model of Wu and Goodman (2018a) and demonstrate that we outperform it at satisfying these four criteria. We evaluate our model on two challenging datasets that capture both image $\leftrightarrow$ image and image $\leftrightarrow$ language transformations, showing appealing results across these tasks.

It is worth noting that the joint learning from multiple modalities, apart from its obvious empirical benefits showcased in our set-up, also holds significant promise in capturing invariant, causally related features in the data. This potential is exemplified by works such as Arjovsky et al. (2019b) and Sturma et al. (2023), with the former highlighting the importance and benefits of identifying invariant factors across different modalities/domains, while the latter work demonstrates that learning shared latent variables across modalities help discover underlying causal relationships. Although these properties of multimodal learning, i.e. improved robustness to distribution shift and learning causally related features, have not been explicitly investigated in this chapter, we envision incorporating them in future research. By harnessing the complementary nature of multiple modalities, such as visual and textual information, our goal is to enhance the robustness, generalisation, and interpretability of models by leveraging invariant, causally linked representations. This direction of exploration holds the potential to uncover valuable insights into the underlying causal mechanisms driving the data generation process.

4

Relating by Contrasting: A Data-efficient Framework for Multimodal DGMs

Abstract

Multimodal learning for generative models often refers to the learning of abstract concepts from the *commonality* of information in multiple modalities, such as vision and language. While it has proven effective for learning generalisable representations, the training of such models often requires a large amount of “related” multimodal data that shares commonality, which can be expensive to come by. To mitigate this, we develop a novel contrastive framework for generative model learning, allowing us to train the model not just by the commonality between modalities, but by the *distinction* between “related” and “unrelated” multimodal data. We show in experiments that our method enables data-efficient multimodal learning on challenging datasets for various multimodal VAE models. We also show that under our proposed framework, the generative model can accurately identify related samples from unrelated ones, making it possible to make use of the plentiful unlabeled, unpaired multimodal data.

4.1 Introduction

To comprehensively describe concepts in the real world, humans collect multiple perceptual signals of the same object such as image, sound, text and video. We refer to each of these media as a *modality*, and a collection of different media featuring the same underlying concept is characterised as *multimodal*

data. Learning from multiple modalities has been

shown to yield more generalisable representations (Zhang et al., 2020; Guo et al., 2019; Yildirim, 2014), as different modalities are often complementary in content while overlapping for common abstract concept.

Despite the motivation, it is worth noting that the multimodal framework is not exactly data-efficient—constructing a suitable dataset requires a lot of “annotated” unimodal data, as we need to ensure that each multimodal pair is related in a meaningful way. The situation is worse when we consider more complicated multimodal settings such as language–vision, where one-to-one or one-to-many correspondence between instances of the two datasets are required, due to the difficulty in categorising data such that commonality amongst samples is preserved within categories. See Figure 4.1 for an example from the CUB dataset (Welinder et al., 2010a); although the same species of bird is featured in both image-caption pairs, their content differs considerably. It would be unreasonable to apply the caption from one to describe the bird depicted in the other, necessitating one-to-one correspondence between images and captions.

However, the scope of multimodal learning has been limited to leveraging the commonality between “related” pairs, while largely ignoring “unrelated” samples potentially available in any multimodal dataset—constructed through random pairing between modalities (fig. 4.3). We posit that if a distinction can be established between the “related” and “unrelated” observations within a multimodal dataset, we could greatly reduce the amount of related data required for effective learning.

Category: Laysan Albatross



"A larger sized bird with a glowing white body and a large orange beak."



"A cigar-shaped white bird with long brown wings and hooked long bill."

Figure 4.1: Multimodal data from the CUB dataset

Figure 4.2 formalises this proposal. Multimodal generative models in previous work (Figure 4.2a) typically assumes one latent variable z that *always* generates related multimodal pair $(\mathbf{x}, \mathbf{y})$. In this work (Figure 4.2b), we introduce an additional Bernoulli random variable r that dictates the “relatedness” between $\mathbf{x}$ and $\mathbf{y}$ through z , where $\mathbf{x}$ and $\mathbf{y}$ are related when $r = 1$, and unrelated when $r = 0$.

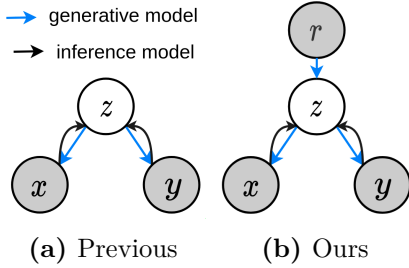


Figure 4.2: Graphical models for multimodal generative process.

While r can encode different dependencies, here we make the simplifying assumption that the Pointwise Mutual Information (PMI) between $\mathbf{x}$ and $\mathbf{y}$ should be high when $r = 1$, and low when $r = 0$. Intuitively, this can be achieved by adopting a max-margin metric. We therefore propose to train the generative models with a novel

contrastive-style loss (Hadsell et al., 2006; Weinberger et al., 2005), and demonstrate the effectiveness of our proposed method from a few different perspectives: **1) Improved multimodal learning:** showing improved multimodal learning for various state-of-the-art multimodal generative models on two challenging multimodal datasets. This is evaluated on four different metrics (Shi et al., 2019a) summarised in Section 4.4.2; **2) Data efficiency:** learning generative models under the contrastive framework requires only 20% of the data needed in baseline methods to achieve similar performance—holding true across different models, datasets and metrics; **3) Label propagation:** the contrastive loss encourages a larger discrepancy between related and unrelated data, making it possible to directly identify related samples using the PMI between observations. We show that these data pairs can be used to further improve the learning of the generative model.

4.2 Related Work

Contrastive loss Our work aims to encourage data-efficient multimodal generative-model learning using a popular *representation learning* metric—contrastive loss (Hadsell et al., 2006; Weinberger et al., 2005). There has been many successful applications of contrastive loss to a range of different tasks, such as contrastive predictive

coding for time series data (van den Oord et al., 2018), image classification (Hénaff, 2020), noise contrastive estimation for vector embeddings of words (Mnih and Kavukcuoglu, 2013), as well as a range of frameworks such as DIM (Hjelm et al., 2019), MoCo (He et al., 2020b), SimCLR (Chen et al., 2020b) for more general visual-representation learning. The features learned by contrastive loss also perform well when applied to different downstream tasks — their ability to generalise is further analysed in (Wang and Isola, 2020) using quantifiable metrics for alignment and uniformity.

While the proposal of incorporating a contrastive objective for multimodal deep generative models in this work was novel, it is important to note that the adaptation of contrastive-style objectives in generative models is not a new concept. Similar techniques can be traced back to earlier approaches, such as the training of energy-based models like the Product of Experts (Hinton, 2002b), which utilised contrastive divergence as a means to minimize the training objective. Additionally, the concept of utilising a contrastive approach is also present in the score function proposed in Hyvärinen and Dayan (2005). Their method aims to estimate statistical models with unknown probability density functions by minimizing the discrepancy between the gradient of the model and the gradient of the observed data. This was achieved by computing the latter through a sample average of a sum of derivatives of the log-density function provided by the model. Contrastive methods have also been employed on GANs to either preserve or identify factors-of-variations in their inputs. For instance, SiGAN (Hsu et al., 2019) uses a contrastive loss to preserve identity for face-image hallucination from low-resolution photos, while (Yildirim et al., 2018) uses a contrastive loss to disentangle the factors of variations in the latent code of GANs. We here employ a contrastive loss in a distinct setting of multimodal generative model learning, that, as we will show with our experiments and analyses promotes better, more robust representation learning.

Multimodal VAEs We also demonstrate that our approach is applicable across different approaches to learning multimodal generative models. To do so, we first summarise past work on multimodal VAE into two categories based on the modelling choice of approximated posterior $q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$:

- a) **Explicit joint models:** q_Φ as single joint encoder $q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$.

Example work in this area include JMVAE (Suzuki et al., 2017), triple ELBO (Vedantam et al., 2018) and MFM (Tsai et al., 2019). Since the joint encoder require multimodal pair $(\mathbf{x}, \mathbf{y})$ as input, these approaches typically require additional modelling components and/or inference steps to deal with missing modality at test time; in fact, all three approaches propose to train unimodal VAEs on top of the joint model that handles data from each modality independently.

- b) **Factorised joint models:** q_Φ as factored encoders $q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = f(q_{\phi_x}(\mathbf{z}|\mathbf{x}), q_{\phi_y}(\mathbf{z}|\mathbf{y}))$.

This was first seen in Wu and Goodman (2018a), as the MVAE model with f defined as a PoE, i.e. $q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = q_{\phi_x}(\mathbf{z}|\mathbf{x})q_{\phi_y}(\mathbf{z}|\mathbf{y})p(\mathbf{z})$, allowing for cross-modality generation without extra modelling components. Particularly, the MVAE caters to settings where data was not guaranteed to be always related, and where additional modalities were, in terms of information content, subsets of a primary data source—such as images and their class labels.

Alternately, Shi et al. (2019a) explored an approach that explicitly leveraged the availability of related/paired data, motivated by arguments from embodied cognition of the world. They propose the MMVAE model, which additionally differs from the MVAE model in its choice of posterior approximation—where f is modelled as the MOE of unimodal posteriors—to ameliorate shortcomings to do with precision miscalibration of the PoE. Furthermore, Shi et al. (2019a) also posit four criteria that a multimodal VAE should satisfy, which we adopt in this work to evaluate the performance of our models.

Weakly-supervised learning Using generative models for label propagation (see Section 4.4.5) is a form of weak supervision. Commonly seen approaches for weakly-supervised training with incomplete data include (1) graph-based methods (such as minimum cut) (Zhou et al., 2003; Zhu et al., 2003; Blum and Chawla, 2001), (2) low-density separation methods (Joachims, 1999; Burkhart and Shan, 2020) and (3) disagreement-based models (Blum and Mitchell, 1998; Zhou and Li, 2005, 2010). However, (1) and (2) suffers from scalability issues due to computational inefficiency and optimisation complexity, while (3) works well for many different tasks but requires training an ensemble of learners.

The use of generative models for weakly supervised learning has also been explored in Nigam et al. (2000); Miller and Uyar (1996), where labels are estimated using expectation-maximisation (EM) (Dempster et al., 1977) for instances that are unlabelled. Notably, models trained with our contrastive objective does not need EM to leverage unlabelled data (see section 4.4.5) to determine the relatedness of two examples we only need to compute a threshold (estimation of PMI) using the trained model.

4.3 Methodology

Given data over observations from two modalities $(\mathbf{x}, \mathbf{y})$, one can learn a multimodal VAE targetting $p_{\Theta}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{z})p_{\theta_x}(\mathbf{x}|\mathbf{z})p_{\theta_y}(\mathbf{y}|\mathbf{z})$, where $p_{\theta}(\cdot|\mathbf{z})$ are deep neural networks (decoders) parametrised by $\Theta = \{\theta_x, \theta_y\}$. To maximise the joint marginal likelihood $\log p_{\Theta}(\mathbf{x}, \mathbf{y})$, one approximates the intractable model posterior $p_{\Theta}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ with a variational posterior $q_{\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})$, allowing us to optimise a variational ELBO, defined as

$$\log p_{\Theta}(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{\mathbf{z} \sim q_{\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})} \left[\log \frac{p_{\Theta}(\mathbf{z}, \mathbf{x}, \mathbf{y})}{q_{\Phi}(\mathbf{z} | \mathbf{x}, \mathbf{y})} \right] = \text{ELBO}(\mathbf{x}, \mathbf{y}). \quad (4.1)$$

This leaves open the question of how to model the approximatd posterior $q_{\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})$. As mentioned in Section 4.2, there are two schools of thinking, namely *explicit joint model* such as JMVAE (Suzuki et al., 2017) and *factorised joint model* including MVAE (Wu and Goodman, 2018a) and MMVAE (Shi et al., 2019a). In this work we demonstrate the effectiveness of our approach across all these models.

4.3.1 Contrastive loss for “relatedness” learning

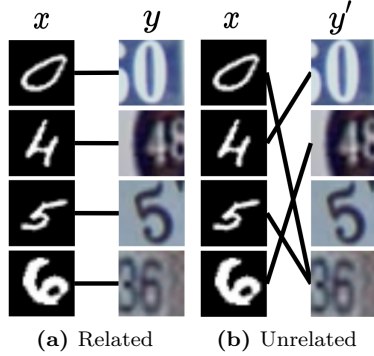


Figure 4.3: Constructing related & unrelated samples

Where prior work always assumes $(\mathbf{x}, \mathbf{y})$ to be related, we introduce a *relatedness* variable r explicitly capturing this aspect. Our approach is motivated by the characteristics of pointwise mutual information (PMI) between related and unrelated observations across modalities:

Hypothesis 4.3.1. *Let $(\mathbf{x}, \mathbf{y}) \sim p_{\Theta}(\mathbf{x}, \mathbf{y})$ be a related data pair from two modalities, and let $\mathbf{y}'$ denote a data point unrelated to $\mathbf{x}$. Then, the pointwise mutual information $I(\mathbf{x}, \mathbf{y}) > I(\mathbf{x}, \mathbf{y}')$.*

Note that the PMI measures the statistical dependence between values (x, y) , which for a joint distribution $p(x, y)$ is defined as $I(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$. Hypothesis 4.3.1 should be a fairly uncontroversial assumption for true generative models: we say simply that under the joint distribution for related data $p_{\Theta}(\mathbf{x}, \mathbf{y})$, the PMI between related points $(\mathbf{x}, \mathbf{y})$ is stronger than that between unrelated points $(\mathbf{x}, \mathbf{y}')$. In fact, we demonstrate in Section 4.4.5 that for a well-trained generative model, PMI is a good indicator of relatedness, enabling pairing of random multimodal data by thresholding the PMI estimated by the trained model. It is therefore possible to leverage relatedness while training parametrised generative model by maximising the difference in PMI between related and unrelated pairs, i.e. $I(\mathbf{x}, \mathbf{y}) - I(\mathbf{x}, \mathbf{y}')$. We show in Appendix C.1 that this is equivalent to maximising the difference between the joint marginal likelihoods of related and unrelated pairs (Figures 4.3a and 4.3b). This casts multimodal learning as max-margin optimisation, with the contrastive (triplet) loss as a natural choice of objective: $\mathcal{L}_C(\mathbf{x}, \mathbf{y}, \mathbf{y}') = d(\mathbf{x}, \mathbf{y}) - d(\mathbf{x}, \mathbf{y}') + m$. Intuitively, $\mathcal{L}_C$ attempts to make distance d between a positive pair $(\mathbf{x}, \mathbf{y})$ smaller than the distance between a negative pair $(\mathbf{x}, \mathbf{y}')$ by margin m . We adopt this loss to our objective by omitting margin m and replacing d with the (negative) joint marginal likelihood $-\log p_{\Theta}$. Following Song et al. (2016), with N negative

samples $\{y'_i\}_{i=1}^N$, we have

$$\mathcal{L}_C(\mathbf{x}, Y) = -\log p_\Theta(\mathbf{x}, \mathbf{y}) + \log \sum_{i=1}^N p_\Theta(\mathbf{x}, \mathbf{y}'_i). \quad (4.2)$$

We choose to put the sum over N within the log following conventions in previous work on contrastive loss (van den Oord et al., 2018; Hjelm et al., 2019; Chen et al., 2020b). As the loss is asymmetric, one can average over $\mathcal{L}_C(\mathbf{y}, X)$ and $\mathcal{L}_C(\mathbf{x}, Y)$ to ensure negative samples in both modalities are accounted for—giving our contrastive objective:

$$\mathcal{L}_C(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \{ \underbrace{\mathcal{L}_C(\mathbf{x}, Y) + \mathcal{L}_C(\mathbf{y}, X)}_{\textcircled{1}} \} = -\log p_\Theta(\mathbf{x}, \mathbf{y}) + \underbrace{\frac{1}{2} \left(\log \sum_{x'_i=1}^N p_\Theta(\mathbf{x}', \mathbf{y}) + \log \sum_{y'_i=1}^N p_\Theta(\mathbf{x}, \mathbf{y}') \right)}_{\textcircled{2}} \quad (4.3)$$

Note that since only the joint marginal likelihood terms are needed in Equation (4.3), the contrastive learning framework can be directly applied to *any* multimodal generative model without needing extra components. We also show our contrastive framework applies when the number of modalities is more than two (cf. Appendix C.3).

Dissecting $\mathcal{L}_C$ Although similar to the VAE Equation (4.1), our objective eq. (4.3) directly maximises $p_\theta(\mathbf{x}, \mathbf{y})$ in $\textcircled{1}$. $\mathcal{L}_C$ by itself is not a effective objective as $\textcircled{2}$ in $\mathcal{L}_C$ *minimises* $p_\theta(\mathbf{x}, \mathbf{y})$,

which can overpower the effect of $\textcircled{1}$ during training, especially when N is large.

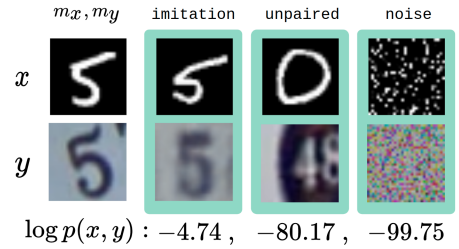


Figure 4.4: $\log p(\mathbf{x}, \mathbf{y})$ of imitation, unrelated digits and random noise, where $p = \mathcal{N}(\mathbf{m}_x, \mathbf{m}_y)$.

We intuit this phenomenon using a simple example in Figure 4.4 with natural images, showing the log likelihood of $\log p(\mathbf{x}, \mathbf{y})$ in column 2, 3, 4 (green) on a Gaussian distribution $\mathcal{N}((\mathbf{m}_x, \mathbf{m}_y); c)$, with the images in the first column of Figure 4.4 as means and with constant variance c . While achieving high $\log p(\mathbf{x}, \mathbf{y})$ requires matching $(\mathbf{m}_x, \mathbf{m}_y)$ (col 2), we see both unrelated digits (col 3) and noise (col 4) can lead to (comparatively) poor joint log likelihoods. This indicates that the generative model need not generate valid, unrelated images to minimise $\textcircled{2}$ —generating noise

would have roughly the same effect on log likelihood. As a result, the model can trivially minimise $\mathcal{L}_C$ by generating noise that minimises ② instead of accurate reconstruction that maximises ①.

This learning dynamic is verified empirically, as we show in Figure C.1 in Appendix C.4: optimising $\mathcal{L}_C$ by itself results in the loss approaching 0 within the first 100 iterations, and both ① and ② takes on extremely low values, resulting in a model that generates random noise.

Final objective To mitigate this issue, we need to ensure that minimising ② does not overpower maximising ①. We hence introduce a hyperparameter γ on ① to upweight the maximisation of the marginal likelihood, with the final objective to minimise

$$\mathcal{L}_C(\mathbf{x}, \mathbf{y}) = \underbrace{-\gamma \log p_\Theta(\mathbf{x}, \mathbf{y})}_{\text{①}} + \underbrace{\frac{1}{2} \left(\log \sum_{\mathbf{x}'=1}^N p_\Theta(\mathbf{x}', \mathbf{y}) + \log \sum_{\mathbf{y}'=1}^N p_\Theta(\mathbf{x}, \mathbf{y}') \right)}_{\text{②}}, \quad \gamma > 1. \quad (4.4)$$

We conduct ablation studies on the effect of γ in Appendix C.5, noting that larger γ encourages better quality of generation and more stable training in some cases, while models trained with smaller γ are better at predicting “relatedness” between multimodal samples. Note that the choice of γ should depend on the number of negative examples N — the true weighting of term ① is $\gamma/\log N$; however, since we fix the value of N in our experiments, we do not make the distinction in our objective for simplicity. We also note that optimising Equation (4.4) maximises the pointwise mutual information $I(\mathbf{x}, \mathbf{y})$; see Appendix C.2 for a proof.

4.3.2 Optimising the objective

Since in VAEs we do not have access to the exact marginal likelihood, we have to optimise an approximated version of the true contrastive objective in Equation (4.4). In Equation (4.5) we list a few possible candidates of estimators and their relationships to the (log) joint marginal likelihood:

$$\text{ELBO} \leq \underbrace{\mathbb{E}_{\{\mathbf{z}_k\}_1^K \sim q_\Phi} \left[\log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(\mathbf{z}_k, \mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{y})} \right]}_{\text{IWAE}} \leq \log p_\theta(\mathbf{x}, \mathbf{y}) \leq \underbrace{\mathbb{E}_{\{\mathbf{z}_k\}_1^K \sim q_\Phi} \left[\log \sqrt{\frac{1}{K} \sum_{k=1}^K \left(\frac{p_\theta(\mathbf{z}_k, \mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{y})} \right)^2} \right]}_{\text{CUBO}} \quad (4.5)$$

with the ELBO from Eq. eq. (4.1), the IWAE, a K -sample lower bound estimator that can compute an arbitrarily tight bound with increasing K (Burda et al., 2015), and the χ upper bound (CUBO), an *upper-bound* estimator (Dieng et al., 2017).

We now discuss our choice of approximation of $\log p_\theta(\mathbf{x}, \mathbf{y})$ for each term in equation Equation (4.4):

- ① Minimising this term maximises the joint marginal likelihood, which can hence be approximated with a valid lower-bound estimator Equation (4.5); the IWAE estimator being the preferred choice.
- ② For this term, we consider two choices of estimators:
 - 1) **CUBO**: Since ② of Equation (4.4) *minimises* the joint marginal likelihoods, to ensure an upper-bound estimator to $\mathcal{L}_C$ in eq. (4.4), we need to employ an upper-bound estimator. Here we propose to use CUBO in Equation (4.5). While such a bounded approximation is indeed desirable, existing upper-bound estimators tend to have rather large bias/variance and can therefore yield poor quality approximations.
 - 2) **IWAE**: We therefore also propose to estimate ② with the IWAE estimator, as it provides an *arbitrarily tight, low variance lower-bound* (Burda et al., 2015) to $\log p_\theta(\mathbf{x}, \mathbf{y})$. Although this no longer ensure a valid bound on the objective, we hope that having a more accurate approximation to the marginal likelihood (and by extension, the contrastive loss) can affect the performance of the model positively.

We report results using both IWAE and CUBO estimators in section 4.4, denoted **cI** and **cC** respectively.

4.4 Experiments

As stated in Section 4.1, we analyse the suitability of contrastive learning for multimodal generative models from three perspectives—*improved multimodal learning* (Section 4.4.3), *data efficiency* (Section 4.4.4) and *label propagation* (Section 4.4.5). Throughout the experiments, we take $N = 5$ negative samples for the contrastive objective, set $\gamma = 2$ based on analyses of ablations in appendix C.5, and take $K = 30$ samples for our IWAE estimators. We now introduce the datasets and metrics used for our experiments.

4.4.1 Datasets

MNIST-SVHN The dataset is designed to separate conceptual complexity, i.e. digit, from perceptual complexity, i.e. color, style, size. Each data pair contains 2 samples of the same digit, one from each dataset (see examples in Figure 4.3a). We construct the dataset such that each instance from one dataset is paired with 30 instances of the same digit from the other dataset. Although both datasets are simple and well-studied, the many-to-many pairing between samples creates matching of different writing styles vs. backgrounds and colors, making it a challenging multimodal dataset.

CUB Image-Captions We also consider a more challenging language-vision multimodal dataset, Caltech-UCSD Birds (CUB) (Welinder et al., 2010b; Reed et al., 2016). The dataset contains 11,788 photos of birds, paired with 10 captions describing the bird’s physical characteristics, collected through Amazon Mechanical Turk (AMT). See CUB image-caption pair in Figure 4.1.

4.4.2 Metrics

Shi et al. (2019a) proposed four criteria for multimodal generative models (Figure 4.5, left), that we summarise and unify as metrics to evaluate these criteria for different generative models (Figure 4.5, right). We now introduce each criterion and its corresponding metric in detail.

(a) **Latent accuracy (Figure 4.5a)** *Criterion: latent space factors into “shared” and “private” subspaces across modalities.* We fit a linear classifier on the samples from $z \sim q_\Phi(z|\mathbf{x}, \mathbf{y})$ to classify the information shared between the two modalities. For MNIST-SVHN, this can be the digit label as shown in Figure 4.5a (right). We check if $\hat{l}_z$ is the same as the digit label of the original inputs $\mathbf{x}$ and $\mathbf{y}$, with the intuition that extracting the commonality between $\mathbf{x}$ and $\mathbf{y}$ from latent representation using a linear transform, supports the claim that the latent space has factored as desired.

(b) **Joint coherence (Figure 4.5b)** *Criterion: model generates paired samples that preserves the commonality observed in data.* Again taking MNIST-SVHN as an example, this can be verified by taking pre-trained MNIST and SVHN digit classifiers and applying them on the multimodal observations generated from the same prior sample z . Coherence is computed by how often generations $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ classify to the same digit, i.e. whether $\hat{l}_x = \hat{l}_y$.

(c) **Cross coherence (Figure 4.5c)** *Criterion: model generates data in one modality conditioned on the other while preserving shared commonality.* To compute cross coherence, we generate observations $\hat{\mathbf{x}}$ using latent from unimodal marginal posterior $z \sim q_\phi(z|\mathbf{y})$ and $\hat{\mathbf{y}}$ from $z \sim q_\phi(z|\mathbf{x})$. Similar to joint coherence, the criterion here is evaluated by predicting the label of the cross-generated samples $\hat{\mathbf{x}}$,

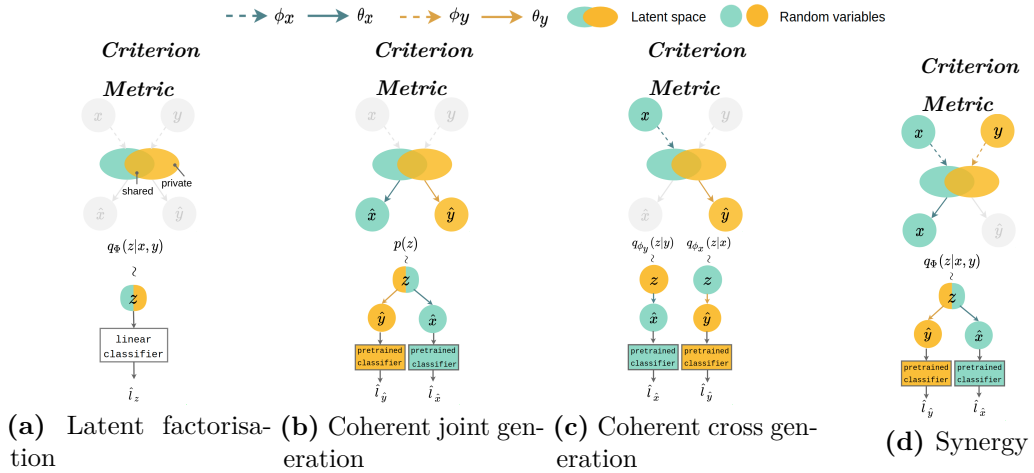


Figure 4.5: Left of each pair: Four criteria for multi-modal generative models; image adapted from Shi et al. (2019a). **Right of each pair:** Four metrics to evaluate the model’s performance on criterion in corresponding row.

$\hat{\mathbf{y}}$ using off-the-shelf MNIST and SVHN classifiers. In Figure 4.5c (right), the cross coherence is the frequency of which $\hat{l}_{\hat{\mathbf{x}}} = l_{\mathbf{y}}$ and $\hat{l}_{\hat{\mathbf{y}}} = l_{\mathbf{x}}$.

(d) Synergy coherence (Figure 4.5d) *Criterion: models learnt across multiple modalities should be no worse than those learnt from just one.* For consistency, we evaluate this criterion from a coherence perspective. Given generations $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ from $\mathbf{z} \sim q_{\Phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})$, we again examine if generated and original labels match; i.e. if $\hat{l}_{\hat{\mathbf{x}}} = l_{\mathbf{y}} = \hat{l}_{\hat{\mathbf{y}}} = l_{\mathbf{x}}$.

See Appendix C.6 for details on architecture. All quantitative results are reported over 5 runs. In addition to these quantitative metrics, we also showcase the qualitative results on both datasets in Appendix C.7 and Appendix C.8.

4.4.3 Improved multimodal learning

Finding: *Contrastive learning improves multimodal learning across all models and datasets.*

MNIST-SVHN See Table 4.1 (*top, 100% data used*) for results on the full MNIST-SVHN dataset. Note that for MMVAE, since the joint posterior q_{Φ} is factorised as the mixture of unimodal posteriors q_{ϕ_x} and q_{ϕ_y} , the model never directly takes sample from the explicit form of the joint posterior. Instead, it takes equal number of samples from each unimodal posteriors, reflective of the equal weighting of the mixture. As a result, it is not meaningful to compute synergy coherence for MMVAE as it is exactly the same as the coherence of any single-way generation.

From Table 4.1 (*top, 100% data used*), we see that models trained on our contrastive objective (cI-<MODEL> and cC-<MODEL>) improves multimodal learning performance significantly for all three generative models evaluated on the metrics. The results showcase the robustness of our approach from the perspectives of modelling choice and metric of interests. In particular, note that for the best performing model MMVAE, using IWAE estimator for ② of Eq Equation (4.4) (cI-MMVAE) yields slightly better results than CUBO (cC-MMVAE), while for the two other models the performance for different estimators are similar.

We also include qualitative results for MNIST-SVHN, including generative results, marginal likelihood table and diversity analysis in Appendix C.7.

Table 4.1: Evaluation of baselines MMVAE, MVAE, JMVAE and their contrastive variations (cI-<MODEL>, cC-<MODEL> for the IWAE and CUBO estimators used in Eq. (4.4), respectively), on MNIST(M)-SVHN(S) dataset, using 100% (top) and 20% (bottom) of data.

Data	Methods	Latent accuracy (%)		Joint Coherence (%)	Cross coherence (%)		Synergy coherence (%)	
		M	S		S→M	M→S	joint→M	joint→S
100% of data used	MMVAE	92.48 (± 0.37)	79.03 (± 1.17)	42.32 (± 2.97)	70.77 (± 0.35)	85.50 (± 1.05)	—	—
	cI-MMVAE	93.97 (± 0.36)	81.87 (± 0.52)	43.94 (± 0.96)	79.66 (± 0.59)	92.67 (± 0.29)	—	—
	cC-MMVAE	93.10 (± 0.17)	80.88 (± 0.80)	45.46 (± 0.78)	79.34 (± 0.54)	92.35 (± 0.46)	—	—
	MVAE	91.65 (± 0.17)	64.12 (± 4.58)	9.42 (± 7.82)	10.98 (± 0.56)	21.88 (± 2.21)	64.60 (± 9.25)	52.91 (± 8.11)
	cI-MVAE	96.97 (± 0.84)	75.94 (± 6.20)	15.23 (± 10.46)	10.85 (± 1.17)	27.70 (± 2.09)	85.07 (± 7.73)	75.67 (± 4.13)
	cC-MVAE	97.42 (± 0.40)	81.07 (± 2.03)	8.85 (± 3.86)	12.83 (± 2.25)	30.03 (± 2.46)	75.25 (± 5.31)	69.42 (± 3.94)
	JMVAE	84.45 (± 0.87)	57.98 (± 1.27)	42.18 (± 1.50)	49.63 (± 1.78)	54.98 (± 3.02)	85.77 (± 0.66)	68.15 (± 1.38)
	cI-JMVAE	84.58 (± 1.49)	64.42 (± 1.42)	48.95 (± 2.31)	58.16 (± 1.83)	70.61 (± 3.13)	93.45 (± 0.52)	84.00 (± 0.97)
	cC-JMVAE	83.67 (± 3.48)	66.64 (± 2.92)	47.27 (± 4.52)	59.73 (± 3.85)	69.49 (± 2.19)	91.21 (± 6.59)	84.07 (± 3.19)
	MMVAE	88.54 (± 0.37)	68.90 (± 1.79)	37.71 (± 0.60)	59.52 (± 0.28)	76.33 (± 2.23)	—	—
	cI-MMVAE	91.64 (± 0.06)	73.02 (± 0.80)	42.74 (± 0.36)	69.51 (± 1.18)	86.75 (± 0.28)	—	—
	cC-MMVAE	92.10 (± 0.19)	71.29 (± 1.05)	40.77 (± 0.93)	68.43 (± 0.90)	86.24 (± 0.89)	—	—
20% of data used	MVAE	90.29 (± 0.57)	33.44 (± 0.26)	10.88 (± 9.15)	8.72 (± 0.92)	12.12 (± 3.38)	42.10 (± 5.22)	44.95 (± 5.92)
	cI-MVAE	93.72 (± 1.09)	56.74 (± 7.97)	12.79 (± 6.82)	14.18 (± 2.19)	20.23 (± 4.55)	75.36 (± 5.05)	64.81 (± 4.81)
	cC-MVAE	92.74 (± 2.97)	52.99 (± 8.35)	17.95 (± 12.52)	14.70 (± 1.65)	24.90 (± 5.77)	56.86 (± 18.84)	54.28 (± 9.86)
	JMVAE	77.53 (± 0.13)	52.55 (± 2.18)	26.37 (± 0.54)	42.58 (± 5.32)	41.44 (± 2.26)	85.07 (± 9.74)	51.95 (± 2.28)
	cI-JMVAE	77.57 (± 4.02)	57.91 (± 1.28)	32.58 (± 5.89)	51.85 (± 1.27)	47.92 (± 10.32)	92.54 (± 1.13)	67.01 (± 8.72)
	cC-JMVAE	81.11 (± 2.76)	57.85 (± 2.23)	34.00 (± 7.18)	50.73 (± 0.45)	56.89 (± 6.18)	88.36 (± 4.38)	68.49 (± 8.82)

CUB Following Shi et al. (2019a), for the images in CUB, we observe and generate in feature space instead of pixel space by preprocessing the images using a pre-trained ResNet-101 (He et al., 2016). A nearest-neighbour lookup among all the features in the test set is used to project the feature generations of the model back to image space. This helps circumvent CUB image complexities to some extent—as the primary goal here is to learn good models and representations of multimodal data, rather than a focus on pixel-level image quality of generations.

The metrics listed in Section 4.4.2 can also be applied to CUB with some modifications. Since bird-species classes are disjoint for the train and test sets, and as we show in Figure 4.1 contains substantial in-class variance, it is not constructive to evaluate these metrics using bird categories as labels. In Shi et al. (2019a), the authors propose to use CCA—used by Massiceti et al. (2018a) as a reliable vision-language correlation baseline—to compute coherence scores between generated image-caption pairs; which we employ (i.e. (b), (c), (d) in Figure 4.5) for CUB.

We show the results in Table 4.2 (*top, 100% data used*). We see that our contrastive approach (both cI-<MODEL> and cC-<MODEL>) is even more effective on this challenging vision-language dataset, with significant improvements to the correlation of generated image-caption pairs. It is also on this more complicated dataset where the advantages of using the stable, low-variance IWAE estimator are highlighted — for both MVAE and JMVAE, the contrastive objective with CUBO estimator suffers from numerical stability issues, yielding close-to-zero correlations for all metrics in Table 4.2. Results for these models are therefore omitted.

We also show the qualitative results for CUB in Appendix C.8.

Table 4.2: Evaluation of baseline MMVAE, MVAE, JMVAE and their contrastive variations (cI-<MODEL> and cC-<MODEL>) on CUB image(img)-caption(cap) dataset, using 100% (top) and 20% (bottom) of data.

Data	Methods	Joint Coherence (CCA)	Cross coherence (CCA)		Synergy coherence (CCA)	
			img→cap	cap→img	joint→img	joint→cap
100% of data used	MMVAE	0.212 ($\pm 2.94e-2$)	0.154 ($\pm 7.05e-3$)	0.244 ($\pm 5.83e-3$)	-	-
	cI-MMVAE	0.314 ($\pm 3.12e-2$)	0.188 ($\pm 4.02e-3$)	0.334 ($\pm 1.20e-2$)	-	-
	cC-MMVAE	0.263 ($\pm 1.47e-2$)	0.244 ($\pm 2.24e-2$)	0.369 ($\pm 3.18e-3$)	-	-
	MVAE	0.075 ($\pm 8.71e-2$)	-0.008 ($\pm 1.41e-4$)	0.000 ($\pm 1.84e-3$)	-0.002 ($\pm 4.95e-3$)	0.001 ($\pm 1.13e-3$)
	cI-MVAE	0.209 ($\pm 3.12e-2$)	0.247 ($\pm 1.12e-4$)	-0.008 ($\pm 3.99e-4$)	0.218 ($\pm 5.27e-3$)	0.148 ($\pm 9.23e-3$)
	JMVAE	0.220 ($\pm 1.19e-2$)	0.157 ($\pm 4.98e-2$)	0.191 ($\pm 3.11e-2$)	0.212 ($\pm 1.23e-2$)	0.143 ($\pm 1.14e-1$)
20% of data used	cI-JMVAE	0.255 ($\pm 3.24e-3$)	0.149 ($\pm 1.25e-3$)	0.226 ($\pm 7.48e-2$)	0.202 ($\pm 1.12e-4$)	0.176 ($\pm 6.23e-2$)
	MMVAE	0.117 ($\pm 1.51e-2$)	0.094 ($\pm 7.21e-3$)	0.153 ($\pm 1.47e-2$)	-	-
	cI-MMVAE	0.206 ($\pm 1.65e-2$)	0.136 ($\pm 1.51e-2$)	0.251 ($\pm 2.39e-2$)	-	-
	cC-MMVAE	0.226 ($\pm 4.69e-2$)	0.188 ($\pm 2.80e-2$)	0.273 ($\pm 8.67e-3$)	-	-
	MVAE	0.091 ($\pm 2.63e-2$)	0.008 ($\pm 4.81e-3$)	0.005 ($\pm 7.50e-3$)	0.020 ($\pm 8.77e-3$)	0.009 ($\pm 1.17e-2$)
	cI-MVAE	0.132 ($\pm 3.33e-2$)	0.192 ($\pm 3.91e-2$)	-0.002 ($\pm 3.61e-3$)	0.162 ($\pm 7.89e-2$)	0.081 ($\pm 3.82e-2$)
	JMVAE	0.127 ($\pm 3.76e-2$)	0.118 ($\pm 3.82e-3$)	0.154 ($\pm 8.34e-3$)	0.181 ($\pm 1.26e-2$)	0.139 ($\pm 1.33e-2$)
	cI-JMVAE	0.269 ($\pm 1.20e-2$)	0.134 ($\pm 4.24e-4$)	0.210 ($\pm 2.35e-2$)	0.192 ($\pm 1.41e-4$)	0.168 ($\pm 3.82e-3$)

4.4.4 Data Efficiency

Finding: *Contrastive learning on 20% of data matches baseline models on full data.*

We plot the quantitative performance of MMVAE with and without contrastive learning against the percentage of the original dataset used, as seen in Figure 4.6. We observe that the performance of contrastive MMVAE with the IWAE (*cI-MMVAE*, *red*) and CUBO estimators (*cC-MMVAE*, *yellow*) are *consistently* better than the baselines (*MMVAE*, *blue*), and that baseline performance using *all* related data

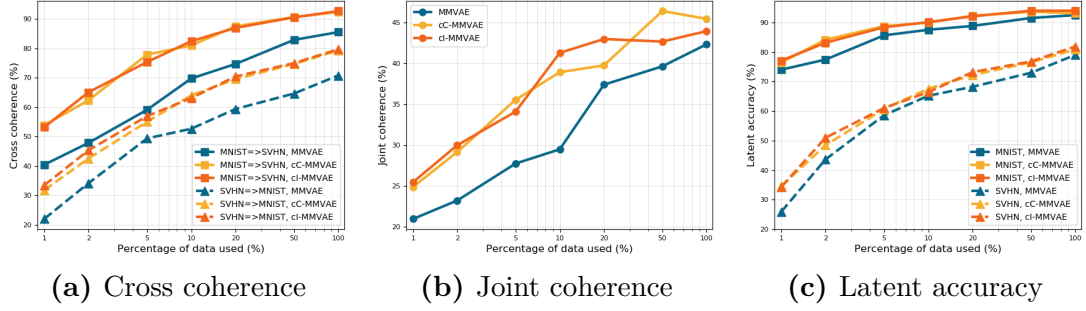


Figure 4.6: Performance of **MMVAE**, **cI-MMVAE** and **cC-MMVAE** using $n\%$ of MNIST-SVHN.

is matched by the contrastive MMVAE using just 10—20% of data. The partial datasets used here are constructed by first taking $n\%$ of each unimodal dataset, then pairing to create multimodal datasets (Section 4.4.1)—ensuring it contains the requisite amount of “related” samples. In addition, we reproduce results generated from using 100% of the data in MNIST-SVHN and CUB (Tables 4.1 and 4.2) using only 20% of the original multimodal datasets (Tables 4.1 and 4.2 (*bottom*)). Comparing results between *top* vs. *bottom* in Tables 4.1 and 4.2 shows that this finding holds across the models, on both MNIST-SVHN and CUB datasets. This shows that the efficiency gains from a contrastive approach is invariant to VAE type, data, and metrics used, underscoring its effectiveness.

4.4.5 Label propagation

Finding: *Generative models learned contrastively are good predictors of “relatedness”, enabling label propagation and matching baseline performance on full datasets, using only 10% of data.*

Here, we show that our contrastive framework encourages a larger discrepancy between the PMI of related vs. unrelated data, as set out in hypothesis 4.3.1, allowing one to first train the model on a small subset of related data, and subsequently construct a classifier using PMI that identifies related samples in the remaining data. We now introduce our pipeline for label propagation in details.

Pipeline As showing in Figure 4.7, we first construct a full dataset by randomly matching instances in MNIST and SVHN, and denote the related pairs by F_r (full, related). We further assume access to only $n\%$ of F_r , denoted as S_r (small,

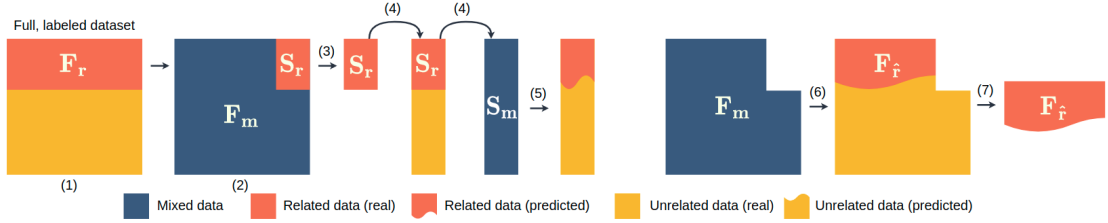
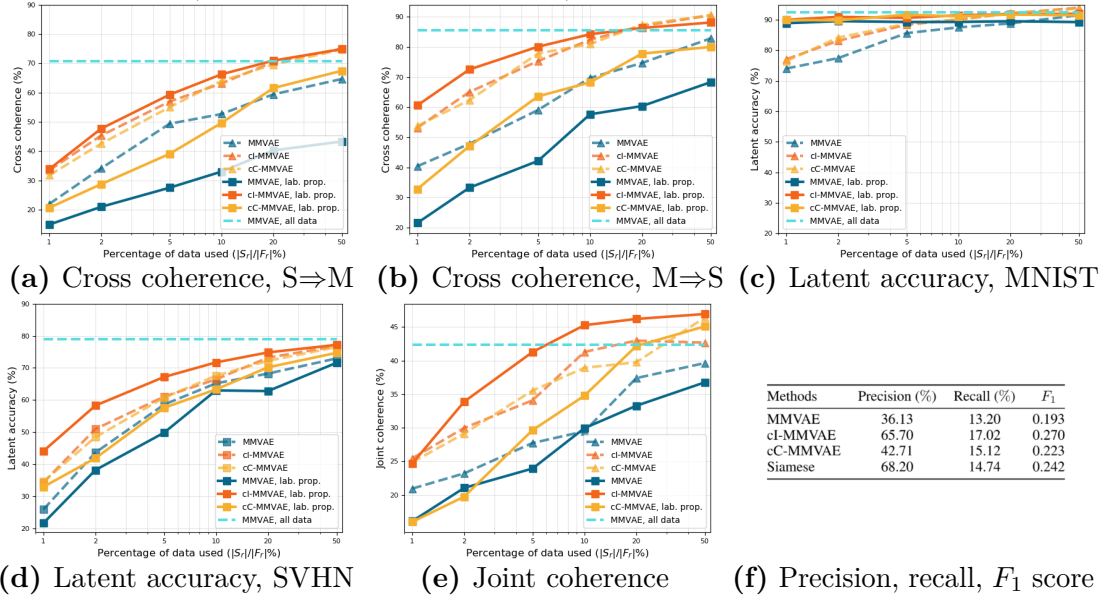


Figure 4.7: Pipeline of label propagation


 Figure 4.8: Models with and without label propagation using **MMVAE**, **cI-MMVAE** and **cC-MMVAE**.

related), and denote the rest as F_m , containing a mix of related and unrelated pairs. Next, we train a generative model g on S_r . To find a relatedness threshold, we construct a small, mixed dataset S_m by randomly matching samples across modalities in S_r . Given relatedness ground-truth for S_m , we can compute the PMI $I(\mathbf{x}, \mathbf{y}) = \log p_{\Theta}(\mathbf{x}, \mathbf{y}) - \log p_{\theta_x}(\mathbf{x})p_{\theta_y}(\mathbf{y})$ for all pairs $(\mathbf{x}, \mathbf{y})$ in S_m and estimate an optimal threshold. This threshold can now be applied to the full, mixed dataset F_m to identify related pairs giving us a new dataset $F_{\hat{r}}$, which can be used to further improve the performance of the generative model g .

Results In Figure 4.8 (a-e), we plot the performance of baseline MMVAE (blue) and contrastive MMVAE with IWAE (cI-MMVAE, red) and CUBO (cC-MMVAE, yellow) estimators for term ② of Eq Equation (4.4), trained with (solid curves) and without (dotted curves) label propagation. Here, the x-axis is the proportion in size of S_r to F_r , i.e. the percentage of related data used to pretrain the generative

model before label propagation. We compare these results to MMVAE trained on all related data F_r (cyan horizontal line) as a “best case scenario” of these training regimes.

Clearly, label propagation using a contrastive model with IWAE estimator is helpful, and in general the improvement is greater when less data is available; Figure 4.8 also shows that when S_r is 10% of F_r , cI-MMVAE is competitive with the performance of baseline MMVAE trained on F_r .

For baselines MMVAE and cC-MMVAE however, label propagation hurts performance no matter the size of S_r , as shown by the blue and yellow curves in Figure 4.8 (a-e). The advantages of cI-MMVAE is further demonstrated in Figure 4.8f, where we compute the precision, recall, and F_1 score of relatedness prediction on F_m , for models trained on 10% of all related data. We also compare to a simple label-propagation baseline, where the relatedness of F_m is predicted using a siamese network (Hadsell et al., 2006) trained on the same 10% dataset. Notably, while the Siamese baseline in Figure 4.8f is a competitive predictor of relatedness, cI-MMVAE has the highest F_1 score amongst all four, making it the most reliable indicator of relatedness. Beyond that, note that with the contrastive MMVAE, relatedness can be predicted *without additional training* and only requires a simple threshold computation directly computed using the generative model. The fact that the cI-MMVAE’s relatedness-prediction performance is the *only* one that matches the Siamese baseline strongly supports the view that the contrastive loss encourages generative models to utilise and better learn the relatedness between multimodal pairs; in addition, the poor performance of cC-MMVAE shows that the advantages of having a bounded estimation to the contrastive objective by using an upper-bound for ② is overshadowed by the high bias of CUBO, and that one may benefit more from choosing a low variance lower-bound like IWAE.

4.5 Conclusion

We introduced a contrastive-style objective for multimodal VAE, aiming at reducing the amount of multimodal data needed by exploiting the distinction between

"related" and "unrelated" multimodal pairs. We showed that this objective improves multimodal training, drastically reduce the amount of multimodal data needed, and establishes a strong sense of "relatedness" for the generative model. These findings hold true across a multitude of datasets, models and metrics. The positive results of our method indicates that it is beneficial to utilise the relatedness information when training on multimodal data, which has been largely ignored in previous work. While we propose to utilise it implicitly through contrastive loss, one may consider relatedness as a random variable in the graphical model and see if explicit dependency on relatedness can be useful. It is also possible to extend this idea to Generative adversarial networks (GANs), by employing an additional discriminator that evaluates relatedness between generations across modalities. We will leave these interesting directions to be explored by future work.

5

Gradient Matching for Domain Generalization

Abstract

Machine learning systems typically assume that the distributions of training and test sets match closely. However, a critical requirement of such systems in the real world is their ability to generalize to unseen domains. Here, we propose an *inter-domain gradient matching* objective that targets domain generalization by maximizing the inner product between gradients from different domains. Since direct optimization of the gradient inner product can be computationally prohibitive — it requires computation of second-order derivatives — we derive a simpler first-order algorithm named Fish that approximates its optimization. We perform experiments on the WILDS benchmark, which captures distribution shift in the real world, as well as the DOMAINBED benchmark that focuses more on synthetic-to-real transfer. Our method produces competitive results on both benchmarks, demonstrating its effectiveness across a wide range of domain generalization tasks. Code is available at <https://github.com/YugeTen/fish>.

5.1 Introduction

The goal of domain generalization is to train models that perform well on unseen, out-of-distribution data, which is crucial in practice for model deployment in the wild. This seemingly

difficult task is made possible by the presence of multiple distri-

butions/domains at train time. Note that in the following discussions, we refer to perceptual characteristics and factors-of-variations in input data as “*features*” (e.g. shape, texture, colors), and the bottleneck output (latent) of the model as “*representations*”.

As we have seen in past work (Arjovsky et al., 2019b; Gulrajani and Lopez-Paz, 2020; Ganin et al., 2016c), a key aspect of domain generalization is to learn from features that remain *invariant* across multiple domains, while ignoring those that are *spuriously correlated* to label information (as defined in Torralba and Efros (2011b); Stock and Cisse (2017))). Consider, for example, a model that is built to distinguish between cows and camels using photos collected in nature under different climates. Since CNNs are known to have a bias towards texture (Geirhos et al., 2018; Brendel and Bethge, 2019), if we simply try to minimize the average loss across different domains, the classifier is prone to spuriously correlate “cow” with grass and “camels” with desert, and predict the species using only the background. Such a classifier can be rendered useless when the animals are placed indoors or in a zoo. However, if the model could recognize that while the landscapes change with climate, the biological characteristics of the animals (e.g. humps, neck lengths) remain invariant and use those features to determine the species, we have a much better chance at generalizing to unseen domains.

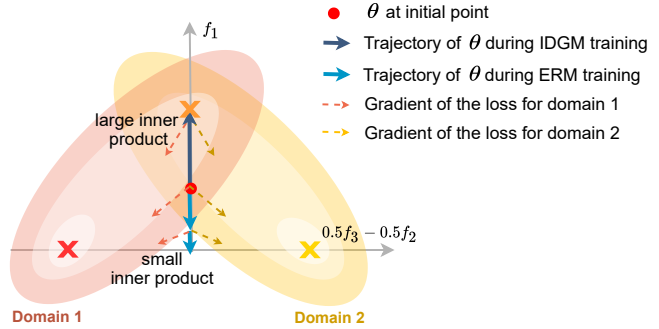


Figure 5.1: Isometric projection of training with ERM (blue) vs. our objective (dark blue), using data from Figure 5.2.

Similar intuitions have already motivated many approaches that consider learning “invariances” across domains as the main challenge of domain generalization. Typically, a lot of these work focus on learning invariant representations, i.e. representations that are completely domain agnostic, directly by removing the domain information (Ganin et al., 2016c; Sun and Saenko, 2016b; Li et al., 2018). In this work, we propose an inter-domain gradient matching (IDGM) objective. Instead of learning invariant features by enforcing invariant representations, our approach does so by encouraging consistent gradient directions across domains. Specifically, our IDGM objective augments the loss with an auxiliary term that maximizes the gradient inner product between domains, which encourages the alignment between the domain-specific gradients. By simultaneously minimizing the loss and matching the gradients, IDGM encourages the optimization paths to be the same for all domains, favouring invariant predictions. Figure 5.1 illustrates a motivating example described in Section 5.3.2: given 2 domains, each containing one invariant feature (orange cross) and one spurious feature (yellow and red cross). While Empirical Risk Minimisation (ERM) minimizes the average loss between these domains at the cost of learning spurious features only, IDGM aligns the gradient directions and is therefore able to focus on the invariant feature.

While the IDGM objective achieves the desirable learning dynamic in theory, naive optimization of the objective by gradient descent is computationally costly due to the second-order derivatives. Leveraging the theoretical analysis of Reptile, a meta-learning algorithm (Nichol et al., 2018), we propose to approximate the gradients of IDGM using a simple first-order algorithm, which we name Fish. Fish is simple to implement, computationally effective and as we show in our experiments, functionally similar to direct optimization of IDGM.

Our contribution is a simple but effective algorithm for domain generalization, which exhibits state-of-the-art performance on 13 datasets from recent domain generalization benchmark WILDS (Koh et al., 2020) and DOMAINBED (Gulrajani and Lopez-Paz, 2020). The strong performance of our method on a variety of datasets

demonstrates that it is broadly applicable in different applications/subgenres of domain generalization tasks. We also perform detailed analysis in Section 5.4.4 to explain the effectiveness of our proposed algorithm.

5.2 Related Work

Domain Generalization In domain generalization, the training data is sampled from one or many source domains, while the test data is sampled from a new target domain. In contrast to domain *adaptation*, the learner does not have access to any data from the target domain (labeled or unlabeled) during train time (Quionero-Candela et al., 2009). In this paper we are interested in the scenario where multiple source domains are available, and the domain where the data comes from is known. Further, Koh et al. (2020) defines the datasets where train and test has disjoint domains “domain generalization”, and those where domains overlap between splits (but typically have different distributions) “subpopulation shift”. Following e.g., Gulrajani and Lopez-Paz (2020), in this paper we use domain generalization in a broader sense that encompasses the two categories.

We will now discuss the five main families of approaches to domain generalization:

1. **Distributional Robustness (DRO)**: DRO approaches minimize the worst-case loss over a set of data distributions constructed from the training domains. Rojas-Carulla et al. (2015) proposed DRO to address *covariate shift* (Gretton et al., 2009a,b), where $P(Y|X)$ remains constant across domains but $P(X)$ changes. Later work also studied *subpopulation shift*, where the train and test distributions are mixtures of the same domains, but the mixture weights change between train and test (Hu et al., 2018; Sagawa et al., 2019);
2. **Domain-invariant representation learning**: This family of approaches to domain generalization aims at learning high-level features that make domains statistically indistinguishable. Prediction is then based on these features only. The principle is motivated by a generalization error bound for unsupervised domain adaptation (Ben-David et al., 2010; Ganin et al., 2016c), but the

approach readily applies to domain generalization (Gulrajani and Lopez-Paz, 2020; Koh et al., 2020). Algorithms include penalising the domain-predictive power of the model (Ganin et al., 2016c; Wang et al., 2019; Huang et al., 2020), aligning domains through contrastive loss (Motiian et al., 2017), matching mean and variance of feature distributions across domains (Sun and Saenko, 2016b), learning useful representations by solving Jigsaw puzzles (Carlucci et al., 2019), using the maximum mean discrepancy to match the feature distributions (Li et al., 2018b) or introducing training constraints across domains using mixup formulation (Yan et al., 2020).

3. **Invariant Risk Minimization (IRM):** IRM is proposed by Arjovsky et al. (2019b), which learns an intermediate representation such that the optimal classifiers (on top of this representation) of all domains are the same. The motivation is to exploit invariant causal effects between domains while reducing the effect of domain-specific spurious correlations. From an optimization perspective, when IRM reaches its optimal, all the gradients (for the linear classifier) *has to be zero*. This is why IRM’s solution won’t deviate from ERM when ERM is optimal for every domain, which is not the case for our proposed IDGM objective due to the gradient inner product term.
4. **Data augmentation:** More recently, approaches that simulates unseen domains through specific types of data augmentation/normalization has been gaining traction. This includes work such as Zhou et al. (2020); Volpi and Murino (2019); Ilse et al. (2021), as well as Seo et al. (2019) which utilises ensemble learning.
5. **Gradient alignment:** Two concurrent work – Koyama and Yamaguchi (2021a) and Parascandolo et al. (2021) – utilise similar gradient-alignment principle for domain generalization. Koyama and Yamaguchi (2021a) proposes IGA, which learns invariant features by minimizing the *variance* of inter-domain gradients. The key difference between IGA and our objective is that IGA is completely identical to ERM when ERM is the optimal solution on

every training domain, since the variances of the gradients will be zero. While they achieve the best performance on the training set, both IGA and ERM *could* in some cases, completely fail when generalizing to unseen domains (see Section 5.3.2 for such an example). Our method, on the contrary, biases towards non-ERM solutions as long as the gradients are aligned, and is therefore able to avoid this issue. Parascandolo et al. (2021) on the other hand, proposes to mask out the gradients that have opposite signs for different domains. Unlike their work that prunes gradients that are inconsistent, our approach actively encourage gradients from different domains to be consistent by maximizing the gradient inner product. Additionally, in Lopez-Paz and Ranzato (2017) we also see the application of gradient-alignment, however in this case it is applied under the continual learning setting to determine whether a gradient update will increase the loss of the previous tasks.

Apart from these algorithms that are tailored for domain generalization, a well-studied baseline in this area is ERM, which simply minimizes the average loss over training domains. Using vanilla ERM is theoretically unfounded (Hashimoto et al., 2018; Blodgett et al., 2016; Tatman, 2017) since ERM is guaranteed to work only when train and test distributions match. Nonetheless, recent benchmarks suggest that ERM obtains strong performance in practice, in many case surpassing domain generalization algorithms (Gulrajani and Lopez-Paz, 2020; Koh et al., 2020). Our goal is to fill this gap, using an algorithm significantly simpler than previous approaches.

Connections to meta-learning There are close connections between meta-learning (Thrun and Pratt, 1998) and (multi-source) domain adaptation. In fact, there are a few works in domain generalization that are inspired by the meta-learning principles, such as Li et al. (2018a); Balaji et al. (2018); Li et al. (2019); Dou et al. (2019). In Ren et al. (2018), we also see the leveraging of gradient inner product in meta-learning, where it is used to determine the importance weight of training examples. We discuss the connection between our proposed algorithm to meta-learning in more details in Appendix D.1.1.

Note that our proposed algorithm Fish is similar to the Mean Teacher method (Tarvainen and Valpola, 2017), where a teacher model (equivalent to θ in Algorithm 1) is computed using a moving average of the student model (equivalent to $\tilde{\theta}$ in Algorithm 1). However with Fish, the "student model" is replaced by the "teacher model" at the beginning of each inner-loop, whereas with Mean Teacher the two models are kept separated.

5.3 Methodology

5.3.1 Goals

Consider a training dataset $\mathcal{D}_{tr}$ consisting of S domains $\mathcal{D}_{tr} = \{\mathcal{D}_1, \dots, \mathcal{D}_S\}$, where each domain s is characterized by a dataset $\mathcal{D}_s := \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ containing data drawn i.i.d. from some probability distribution. Also consider a test dataset $\mathcal{D}_{te}$ consisting of T domains $\mathcal{D}_{te} = \{\mathcal{D}_{S+1}, \dots, \mathcal{D}_{S+T}\}$, where $\mathcal{D}_{tr} \cap \mathcal{D}_{te} = \emptyset$. The goal of domain generalization is to train a model with weights θ that generalizes well on the test dataset $\mathcal{D}_{te}$ such that:

$$\arg \min_{\theta} \mathbb{E}_{\mathcal{D} \sim \mathcal{D}_{te}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [l((x, y); \theta)], \quad (5.1)$$

where $l((x, y); \theta)$ is the loss of model θ evaluated on (x, y) .

A naive approach is to apply ERM, which simply minimizes the average loss on $\mathcal{D}_{tr}$, ignoring the discrepancy between train and test domains:

$$\mathcal{L}_{\text{erm}}(\mathcal{D}_{tr}; \theta) = \mathbb{E}_{\mathcal{D} \sim \mathcal{D}_{tr}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [l((x, y); \theta)]. \quad (5.2)$$

The ERM objective does not exploit the invariance across different domains in $\mathcal{D}_{tr}$ and could perform arbitrarily poorly on test data. We demonstrate this effect with the following simple linear example.

5.3.2 The pitfall of ERM: a linear example

Consider a binary classification setup where data $(x, y) \in \mathbb{B}^4 \times \mathbb{B}$, and a data instance is denoted $x = [f_1, f_2, f_3, f_4], y$. The train domains are $\{\mathcal{D}_1, \mathcal{D}_2\}$, and test domain is $\mathcal{D}_3$. The goal is to learn a linear model $Wx + b = y, W \in \mathbb{R}^4, b \in \mathbb{R}$ on

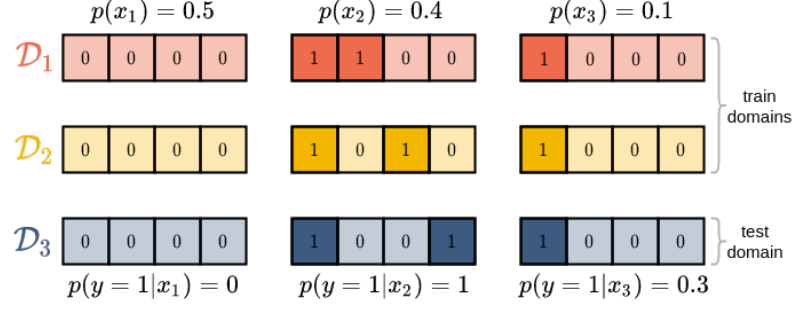


Figure 5.2: All 3 domains (rows) consist of 3 types of inputs (columns): 1) x_1 , **left:** makes up for 50% of each domain, label is always 0, x_1 is always $[0, 0, 0, 0]$; 2) x_2 , **middle:** makes up for 40% of each domain, label is always 1, x_2 changes for each domain; 3) x_3 , **right:** makes up for 10% of each domain, labels are randomly assigned with 30% of $y = 1$ and 70% of $y = 0$, x_3 is always $[1, 0, 0, 0]$.

the train data, such that the error on the test domain is minimized. The setup and dataset of this example is illustrated in Figure 5.2.

As we can see in Figure 5.2, f_1 is the *invariant feature* in this dataset, since the correlation between f_1 and y is stable across different domains. The relationships between y and f_2 , f_3 and f_4 changes for $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, making them the *spurious features*. Importantly, if we consider one domain only, the spurious features f_2 , f_3 and f_4 are a more accurate indicator of the label than the invariant feature f_1 . For instance, using f_2 to predict y can give 97% accuracy on $\mathcal{D}_1$, while using f_1 only achieves 93% accuracy.

Table 5.1: Performance comparison on the linear dataset.

Method	train acc.	test acc.	W	b
ERM	97%	57%	$[2.8, 3.3, 3.3, 0.0]$	-2.7
IDGM	93%	93%	$[0.4, 0.2, 0.2, 0.0]$	-0.4
Fish	93%	93%	$[0.4, 0.2, 0.2, 0.0]$	-0.4

The performance of ERM on this simple example is shown in Table 5.1 (first row). From the trained parameters W and b , we see that the model places most of its weights on spurious features f_2 and f_3 . While this achieves the highest train accuracy (97%), the model cannot generalize to unseen domains and performs poorly on test accuracy (57%).

5.3.3 Inter-domain Gradient Matching (IDGM)

To mitigate the problem with ERM, we need an objective that learns from features that are invariant across domains. Let us consider the case where the train dataset consists of $S = 2$ domains $\mathcal{D}_{tr} = \{\mathcal{D}_1, \mathcal{D}_2\}$. Given model θ and loss function l , the expected gradients for data in the two domains is expressed as

$$G_1 = \mathbb{E}_{\mathcal{D}_1} \frac{\partial l((x, y); \theta)}{\partial \theta}, \quad G_2 = \mathbb{E}_{\mathcal{D}_2} \frac{\partial l((x, y); \theta)}{\partial \theta}. \quad (5.3)$$

The direction, and by extension, inner product of these gradients are of particular importance to our goal of learning invariant features. If G_1 and G_2 point in a similar direction, i.e. $G_1 \cdot G_2 > 0$, taking a gradient step along G_1 or G_2 improves the model’s performance on both domains, indicating that the features learned by either gradient step are invariant across $\{\mathcal{D}_1, \mathcal{D}_2\}$. This invariance cannot be guaranteed if G_1 and G_2 are pointing in opposite directions, i.e. $G_1 \cdot G_2 \leq 0$.

To exploit this observation, we propose to maximize the gradient inner product (GIP) to align the gradient direction across domains. The intended effect is to find weights such that the input-output correspondence is as close as possible across domains. We name our objective *inter-domain gradient matching* (IDGM), and it is formed by subtracting the inner product of gradients between domains $\hat{G}$ from the original ERM objective. For the general case where $S \geq 2$, we can write

$$\mathcal{L}_{\text{idgm}} = \mathcal{L}_{\text{erm}}(\mathcal{D}_{tr}; \theta) - \underbrace{\gamma \frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} G_i \cdot G_j}_{\text{GIP, denote as } \hat{G}}, \quad (5.4)$$

where γ is the scaling term for $\hat{G}$. Note that GIP can be computed in linear time as $\hat{G} = \|\sum_i G_i\|^2 - \sum_i \|G_i\|^2$ (ignoring the constant factor). We can also compute the stochastic estimates of Equation (5.4) by replacing out the expectations over the entire dataset by minibatches.

We test this objective on our simple linear dataset, and report results in the second row of Table 5.1. Note that to avoid exploding gradient we use the normalized GIP during training. The model has lower training accuracy compared to ERM (93%),

however its accuracy remains the same on the test set, much higher than ERM. The trained weights W reveal that the model assigns the largest weight to the invariant feature f_1 , which is desirable. The visualization in Figure 5.1 also confirms that by maximizing the gradient inner product, IDGM is able to focus on the feature that is common between domains, yielding better generalization performance than ERM.

5.3.4 Optimizing IDGM with Fish

The proposed IDGM objective, although effective, requires computing the second-order derivative of the model’s parameters due to the gradient inner product term, which can be computationally prohibitive. To mitigate this, we propose a first-order algorithm named Fish¹ that approximates the optimization of IDGM with inner-loop updates. In Algorithm 1 we present Fish. As a comparison, we also present direct optimization of IDGM using SGD in Algorithm 2.

Algorithm 1: Fish.

```

1: for iterations = 1, 2, ... do
2:    $\tilde{\theta} \leftarrow \theta$ 
3:   for  $\mathcal{D}_i \in \text{permute}(\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_S\})$  do
4:     Sample batch  $d_i \sim \mathcal{D}_i$ 
5:      $\tilde{g}_i = \mathbb{E}_{d_i} \left[ \frac{\partial l((x, y); \tilde{\theta})}{\partial \tilde{\theta}} \right]$  // Grad wrt  $\tilde{\theta}$ 
6:     Update  $\theta \leftarrow \tilde{\theta} - \alpha \tilde{g}_i$ 
7:   end for
8:   Update  $\theta \leftarrow \theta + \epsilon(\tilde{\theta} - \theta)$ 
9: end for
```

Algorithm 2: Direct optimization of IDGM.

```

1: for iterations = 1, 2, ... do
2:    $\tilde{\theta} \leftarrow \theta$ 
3:   for  $\mathcal{D}_i \in \text{permute}(\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_S\})$  do
4:     Sample batch  $d_i \sim \mathcal{D}_i$ 
5:      $g_i = \mathbb{E}_{d_i} \left[ \frac{\partial l((x, y); \theta)}{\partial \theta} \right]$  // Grad wrt  $\theta$ 
6:   end for
7:    $\bar{g} = \frac{1}{S} \sum_{s=1}^S g_s$ ,  $\hat{g} = \frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} g_i \cdot g_j$ 
8:   Update  $\theta \leftarrow \theta - \epsilon(\bar{g} - \gamma(\partial \hat{g} / \partial \theta))$ 
9: end for
```

Fish performs S inner-loop (*l3-l7*) update steps with learning rate α on a clone of the original model $\tilde{\theta}$, and each update uses a minibatch d_i from the domain selected in step i . Subsequently, θ is updated by a weighted difference between the cloned model and the original model $\epsilon(\tilde{\theta} - \theta)$.

To see why Fish is an approximation to directly optimizing IDGM, we can perform Taylor-series expansion on its update in *l8*, Algorithm 1. Doing so reveals two leading terms: 1) $\bar{g}$: averaged gradients over inner-loop’s minibatches (effectively

¹Following the convention of naming this style of algorithms after classes of vertebrates (animals with backbones).

the ERM gradient); 2) $\partial\hat{g}/\partial\theta$: gradient of the minibatch version of GIP. Observing *l8* of Algorithm 2, we see that $\bar{g}$ and $\hat{g}$ are actually the two gradient components used in direct optimization of IDGM. Therefore, Fish implicitly optimizes IDGM by construction (up to a constant factor), avoiding the computation of second-order derivative $\partial\hat{g}/\partial\theta$. We present this more formally for the full gradient G in Theorem 5.3.1.

Theorem 5.3.1. *Given twice-differentiable model with parameters θ and objective l . Let us define the following:*

$$\begin{aligned} G_f &= \mathbb{E}[(\theta - \tilde{\theta})] - \alpha S \cdot \bar{G}, & \text{Fish update - } \alpha S \cdot \text{ERM grad} \\ G_g &= -\partial\hat{G}/\partial\theta, & \text{grad of } \max_{\theta}(\hat{G}) \end{aligned}$$

where α is the learning rate, and $\bar{G} = \frac{1}{S} \sum_{s=1}^S G_s$ is the full gradient of ERM. Then we have

$$\lim_{\alpha \rightarrow 0} \frac{G_f \cdot G_g}{\|G_f\| \cdot \|G_g\|} = 1.$$

Note that the expectation in G_f is over the sampling of domains and minibatches. Theorem 5.3.1 indicates that when the learning rate α is sufficiently small, if we remove the scaled ERM gradient component $\bar{G}$ from Fish's update, we are left with a term G_f that is in similar direction to the gradient of maximizing the GIP term in IDGM, which was originally second-order. To see this, let us expand both G_g and G_f .

Expanding G_g G_g is the gradient of maximizing the gradient inner product (GIP).

$$G_g = -\frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial}{\partial \theta} G_i \cdot G_j \quad (5.5)$$

Expanding G_f To write out G_f , we need to first derive the gradient update of Fish, $\theta - \tilde{\theta}$. By applying Taylor approximation to the gradient of each step of Fish

update ($S = 1$), we can write the expected update of the algorithm where $S > 1$ as (see Appendix D.1 for more details)

$$\mathbb{E}[\theta - \tilde{\theta}] = \alpha \sum_{s=1}^S G_s - \frac{\alpha^2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial(G_i \cdot G_j)}{\partial \theta_1} + \mathcal{O}(\alpha^3). \quad (5.6)$$

The second term of G_f in Theorem 5.3.1 is $\bar{G}$, which is the full gradient of ERM defined as follow:

$$\bar{G} = \frac{1}{S} \sum_{s=1}^S G_s. \quad (5.7)$$

Plugging Equation (5.6) and Equation (5.7) to G_f defined in Theorem 5.3.1 yields

$$G_f = \mathbb{E}[\theta - \tilde{\theta}] - \alpha S \bar{G} = -\frac{\alpha^2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial}{\partial \theta_1} G_i \cdot G_j \quad (5.8)$$

Comparing Equation (5.5) to Equation (5.8), we have:

$$\lim_{\alpha \rightarrow 0} \frac{G_f \cdot G_g}{\|G_f\| \|G_g\|} = 1.$$

Note that the approximation of Fish comes at the cost of losing direct control over the GIP scaling γ — we therefore also derived a smoothed version of Fish that recovers this scaling term, however we find that changing the value of γ does not make much difference empirically. See Appendix D.2 for more details.

The proof to Theorem 5.3.1 can be found in Appendix D.1. We follow the analysis from Nichol et al. (2018), which proposes Reptile for model-agnostic meta-learning (MAML), where the relationship between inner-loop update and maximization of gradient inner product was first highlighted. Nichol et al. (2018) found the GIP term in their algorithm to be over minibatches from the *same domain*, which promoted within-task generalization; in Fish we construct inner-loop using minibatches over *different domains* — it therefore instead encourages across-domain generalization. We compare the two algorithms in further details in Appendix D.1.1.

We also train Fish on our simple linear dataset, with results in Table 5.1, and see it performs similarly to IDGM — the model assigns the most weight to the invariant feature f_1 , and achieves 93% accuracy on both train and test dataset.

5.4 Experiments

5.4.1 CdSprites-N

Dataset We propose a simple shape-color dataset CDSPRITES-N based on the DSPRITES dataset (Matthey et al., 2017), which contains a collection of white 2D sprites of different shapes, scales, rotations and positions. CDSPRITES-N contains N domains. The goal is to classify the shape of the sprites, and there is a shape-color deterministic matching that is specific per domain. This way we have shape as the invariant feature and color as the spurious feature. See Figure 5.3 for an illustration.

To construct the train split of CDSPRITES-N, we take a subset of DSPRITES that contains only 2 shapes (square and oval). We make N replicas of this subset and assign 2 colors to each, with every color corresponding to one shape (e.g. yellow block in Figure 5.3a, pink $\rightarrow$ squares, purple $\rightarrow$ oval). For the test split, we create another replica of the DSPRITES-N subset, and randomly assign one of the $2N$ colors in the training set to each shape in the test set.

We design this dataset with CNN’s texture bias in mind (Geirhos et al., 2018; Brendel and Bethge, 2019). If the value of N is small enough, the model can simply memorize the N colors that correspond to each shape, and make predictions solely based on colors, resulting in poor performance on the test set where color and shape are no longer correlated. Our dataset allows for precise control over the features that remains stable across domains and the features that change as domains change; we can also change the number of domains N easily, making it possible to examine the effect N has on the performance for domain generalization.

Results We train the same model using three different objectives including Fish, direct optimization of IDGM and ERM on this dataset with number of domains N ranging from 5 to 50. We train all models using a simple 5-layer CNN. Details of the architecture can be found in Appendix D.3. Again, for direct optimization of IDGM, we use the normalized gradient inner product to avoid exploding gradient.

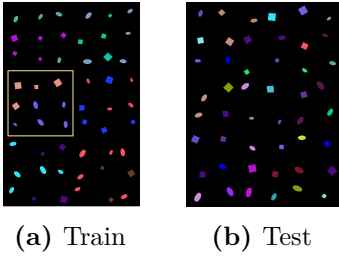


Figure 5.3: CDSprites-N visualization. Each 3x3 grid (e.g. yellow square) is one domain.

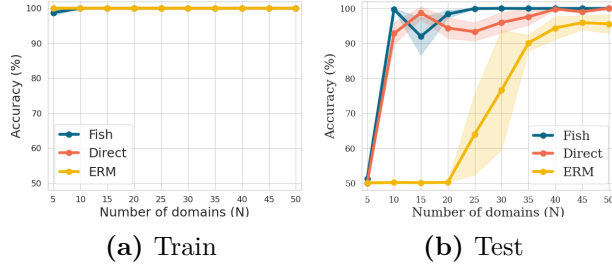


Figure 5.4: Performance of Fish, IDGM and ERM on CDSprites-N, with $N \in [5, 50]$

We plot the average train, test accuracy for each objective over 5 runs against the number of domains N in Figure 5.4. We can see that the train accuracy is always 100% for all methods regardless of N (Figure 5.4a), while the test performance varies: Figure 5.4b shows that **direct** optimization of IDGM (red) and **Fish** (blue) obtain the best performances, with the test accuracy rising to over 90% when $N \geq 10$ and near 100% when $N \geq 20$. The predictions of **ERM** (yellow), on the other hand, remain nearly random on the test set up until $N = 20$, and reach 95% accuracy only for $N \geq 40$.

This experiment confirms the following: 1) the proposed IDGM objective have much stronger domain generalization capabilities compared to ERM; 2) Fish is an effective approximation of IDGM, with similar performance to its direct optimization. We also plot the gradient inner product progression of Fish vs. ERM during training in Figure D.4a, showing clearly that Fish does improve the gradient inner product across domain while ERM does not; 3) we also observe during training that Fish is about 10 times faster than directly optimizing IDGM, demonstrating its computational efficiency.

5.4.2 Wilds

Datasets We evaluate our model on the WILDS benchmark (Koh et al., 2020), which contains multiple datasets that capture real-world distribution shifts across a diverse range of modalities. We report experimental results on 6 challenging datasets in WILDS, and find Fish to outperform all baselines on most tasks. A summary of the WILDS datasets can be found in Appendix D.4. For hyperparameters including learning rate, batch size, choice of optimizer and model architecture, we follow the

exact configuration as reported in the WILDS benchmark. Importantly, we also use the same model selection strategy used in WILDS to ensure a fair comparison. See details in Appendix D.5.

Table 5.2: Results on WILDS benchmark.

	POVERTYMAP	CAMELYON17	FMoW	CIVILCOMMENTS	iWILDCAM	AMAZON
	Worst-U/R Pearson r	Avg. acc. (%)	Worst acc. (%)	Worst acc. (%)	Macro F1	10-th per. acc. (%)
Fish	0.30 ($\pm 1e-2$)	74.7 (± 7.1)	34.6 (± 0.18)	75.3 (± 0.6)	22.0 (± 0.0)	53.3 (± 0.0)
IRM	0.43 ($\pm 7e-2$)	64.2 (± 8.1)	30.0 (± 1.37)	66.3 (± 2.1)	15.1 (± 4.9)	52.4 (± 0.8)
Coral	0.44 ($\pm 6e-2$)	59.5 (± 7.7)	31.7 (± 1.24)	65.6 (± 1.3)	32.8 (± 0.1)	52.9 (± 0.8)
Rewighted	-	-	-	69.2 (± 0.9)	-	52.0 (± 0.0)
GroupDRO	0.39 ($\pm 6e-2$)	68.4 (± 7.3)	30.8 (± 0.81)	70.0 (± 2.0)	23.9 (± 2.1)	53.3 (± 0.0)
ERM	0.45 ($\pm 6e-2$)	70.3 (± 6.4)	32.3 (± 1.25)	56.0 (± 3.6)	31.0 (± 1.3)	53.8 (± 0.8)
ERM (ours)	0.29 ($\pm 1e-2$)	70.5 (± 12.1)	30.9 (± 1.53)	58.1 (± 1.7)	25.1 (± 0.2)	53.3 (± 0.8)

Results See a summary of results in Table 5.2, where we use the metrics recommended in WILDS for each dataset. Again, following practices in WILDS, all results are reported over 3 random seed runs, apart from CAMELYON17 which uses 10 random seeds and CIVILCOMMENTS which uses 5. We included additional results as well as a in-depth discussion on each dataset in Appendix D.4, and an ablation studies on Fish’s hyperparameters in Appendix D.7 and Appendix D.6. We make the following observations:

1. **Strong performance across datasets:** Considering results on all 6 datasets, Fish is the best performing algorithm on WILDS. It significantly outperforms baselines on 3 datasets and achieves similar level of performance to the best method on the other 3 (AMAZON and iWILDCAM). Fish’s strong performance on different types of data and architectures such as RESNET (He et al., 2016), DENSENET (Huang et al., 2017) and DISTILBERT (Sanh et al., 2019) demonstrated it’s capability to generalize to a diverse variety of tasks;
2. **Strong performance on different domain generalization tasks:** We make special note the CIVILCOMMENTS dataset captures *subpopulation shift* problems, where the domains in test are a subpopulation of the domains in train, while all other WILDS datasets depicts *pure domain generalization* problems, where the domains in train and test are disjointed. As a result, the baseline models for CIVILCOMMENTS selected by the WILDS benchmark

Table 5.3: Test accuracy (%) on DOMAINBED benchmark.

	ERM	IRM	GroupDRO	Mixup	MLDG	Coral	MMD	DANN	CDANN	Fish (ours)
CMNIST	51.5 (± 0.1)	52.0 (± 0.1)	52.1 (± 0.0)	52.1 (± 0.2)	51.5 (± 0.1)	51.5 (± 0.1)	51.5 (± 0.2)	51.5 (± 0.3)	51.7 (± 0.1)	51.6 (± 0.1)
RMNIST	98.0 (± 0.0)	97.7 (± 0.1)	98.0 (± 0.0)	98.0 (± 0.1)	97.9 (± 0.0)	98.0 (± 0.1)	97.9 (± 0.0)	97.8 (± 0.1)	97.9 (± 0.1)	98.0 (± 0.0)
VLCS	77.5 (± 0.4)	78.5 (± 0.5)	76.7 (± 0.6)	77.4 (± 0.6)	77.2 (± 0.4)	78.8 (± 0.6)	77.5 (± 0.9)	78.6 (± 0.4)	77.5 (± 0.1)	77.8 (± 0.3)
PACS	85.5 (± 0.2)	83.5 (± 0.8)	84.4 (± 0.8)	84.6 (± 0.6)	84.9 (± 1.0)	86.2 (± 0.3)	84.6 (± 0.5)	83.6 (± 0.4)	82.6 (± 0.9)	85.5 (± 0.3)
OfficeHome	66.5 (± 0.3)	64.3 (± 2.2)	66.0 (± 0.7)	68.1 (± 0.3)	66.8 (± 0.6)	68.7 (± 0.3)	66.3 (± 0.1)	65.9 (± 0.6)	65.8 (± 1.3)	68.6 (± 0.4)
TerraInc	46.1 (± 1.8)	47.6 (± 0.8)	43.2 (± 1.1)	47.9 (± 0.8)	47.7 (± 0.9)	47.6 (± 1.0)	42.2 (± 1.6)	46.7 (± 0.5)	45.8 (± 1.6)	45.1 (± 1.3)
DomainNet	40.9 (± 0.1)	33.9 (± 2.8)	33.3 (± 0.2)	39.2 (± 0.1)	41.2 (± 0.1)	41.5 (± 0.1)	23.4 (± 9.5)	38.3 (± 0.1)	38.3 (± 0.3)	42.7 (± 0.2)
Average	66.6	65.4	64.8	66.7	66.7	67.5	63.3	66.1	65.6	67.1

are different from the methods used in all other datasets, and are tailored to avoiding systematic failure on data from minority subpopulations. We see that Fish works well in this setting too without any changes or special sampling strategies (used for baselines on CIVILCOMMENTS, see more in Table D.7), demonstrating it’s capability to perform in different domain generalization scenarios;

3. **Failure mode of domain generalization algorithms:** We noticed that on IWILDCAM and AMAZON, ERM is the best algorithm, outperforming all domain generalization algorithms except for Fish on AMAZON. We believe that these domain generalization algorithms failed due to the large number of domains in these two datasets — 324 for IWILDCAM and 7,676 for AMAZON. This is a common drawback of current domain generalization literature and is a direction worth exploring.

5.4.3 DomainBed

Datasets While WILDS is a challenging benchmark capturing realistic distribution shift, to test our model under the synthetic-to-real transfer setting and provide more comparisons to SOTA methods, we also performed experiments on the DOMAINBED benchmark (Gulrajani and Lopez-Paz, 2020). See a summary of DOMAINBED in Appendix D.9.

Results Following recommendations in DOMAINBED, we report results using training domain as validation set for model selection. See results in Table 5.3, reported over 5 random trials. Averaging the performance over all 7 datasets, Fish ranks second out of 10 domain generalization methods. It performs only marginally

worse than Coral (0.1%), and is one of the three methods that performs better than ERM. This showcases Fish’s effectiveness on domain generalization datasets with stronger focus to synthetic-to-real transfer, which again demonstrates its versatility and robustness on different domain generalization tasks.

5.4.4 Analysis

We show extensively through empirical evaluation that Fish is very effective for a variety domain generalization tasks. In this section, we perform analysis to validate that Fish’s strong performance is due to inter-domain gradient inner product maximization.

Does Fish maximize gradient inner product (GIP) empirically? In Figure 5.5, we plot the progression of GIP during training using different objectives.

We train both **Fish** (blue) and **ERM** (yellow) on CDSPRITES-N until convergence while tracking the normalized GIP

between minibatches from different domains used in each inner-loop. To ensure a fair comparison, we use the exact same sequence of data for Fish and ERM (see Appendix D.10 for more details).

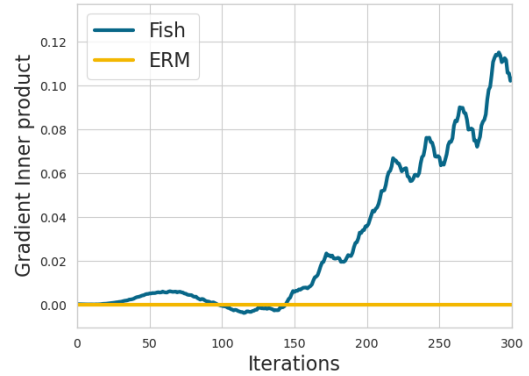


Figure 5.5: Gradient inner product values during the training for CDSPRITES-N (N=15).

From Figure 5.5, it is clear that during training, the normalized GIP of Fish increases, while that for ERM stays at the same value. The observations here shows that Fish is indeed effective in increasing/maintaining the level of inter-domain GIP.

We conduct the same GIP tracking experiments for the WILDS datasets we studied as well to shed some lights on its efficiency — see Appendix D.8 for results.

Does maximizing GIP between random batches help domain generalization? In this part of the analysis, we examine the effectiveness of another element

Table 5.4: Test accuracy on four datasets, in three partitions are 1) two baselines Fish and ERM, 2) Fish with random grouping strategy (*Fish*, *RG*) and 3) ERM with domain grouping strategy (*ERM*, *DG*, $*$)

	FMoW	VLCS	PACS	OfficeHome
Fish (ours)	34.3 (± 0.6)	77.6 (± 0.5)	85.5 (± 0.3)	68.6 (± 0.9)
ERM	31.7 (± 1.0)	77.5 (± 0.4)	85.5 (± 0.2)	66.5 (± 0.3)
Fish, RG	33.4 (± 1.7)	77.7 (± 0.3)	83.9 (± 0.7)	66.5 (± 1.0)
ERM, DG, prop. lr	32.1 (± 0.5)	72.7 (± 0.4)	83.2 (± 0.7)	58.5 (± 0.1)
ERM, DG, $0.1 \times$ prop. lr	29.9 (± 0.7)	73.9 (± 0.6)	84.2 (± 0.4)	65.1 (± 0.1)
ERM, DG, $0.01 \times$ prop. lr	29.8 (± 0.4)	74.7 (± 0.4)	84.0 (± 0.6)	63.7 (± 0.2)

of our algorithm – the construction of minibatches. We conduct experiments where data are grouped randomly instead of by domain for the inner-loop update. By doing so, we are still maximizing the inner product between minibatches, however not strictly between domain. We therefore expect the results to be slightly worse than Fish, and the bigger the domain gap is, the more advantage Fish has against the random grouping strategy.

We show the results for random grouping (*Fish*, *RG*) in Table 5.4. As expected, the random grouping strategy performs worse than Fish on all datasets. The experiment demonstrated that our algorithm does benefit from the domain grouping strategy, and that maximising GIP between random batches of data, while still achieving strong results, does not achieve the same domain generalization performance as Fish.

Differences between Fish and ERM We have shown through empirical evaluation that Fish and ERM are sufficiently different. Most notably, they converge to completely distinct minimum in our linear toy example and CDSprites experiment, resulting in large discrepancies in test accuracy.

However, some may notice that Fish as a first-order method shares algorithmic similarities to ERM. In fact, if we fix the meta learning rate of Fish $\epsilon = 1$ and use domain sampling for ERM, the two algorithms are equivalent. Although the optimal ϵ s found through hyperparameter search are much smaller than 1, one might wonder if we can offset the increased meta learning rate ϵ by reducing the

learning rate α proportionally, and achieve similar performance to Fish using simply ERM with domain grouping.

In Table 5.4 (*ERM, DG*), we verify that this is infeasible empirically. We train an ERM model with domain grouping strategy (equivalent to Fish with $\epsilon = 1$) using 3 different learning rates: 1) *prop. lr*, learning rate α is lowered proportionally to keep $\alpha\epsilon$ constant; 2) $0.1 \times \text{prop. lr}$ and 3) $0.001 \times \text{prop. lr}$ which further reduces the learning rate. We see that on all datasets, all three configurations result in worse performance than ERM and Fish. This shows that one cannot improve the results of ERM to match that of Fish by simply adjusting the learning rate/adopting domain grouping strategy, and that Fish’s gain on domain generalization tasks is not due to its proximity to ERM.

In Appendix D.1.2, we also show that setting $\epsilon = 1$ is equivalent to setting S as the total number of iterations during training, which causes the effect of maximizing GIP to diminish. This provides further support to our empirical finding in this analysis.

Does maximising GIP help domain generalization? In Parascandolo et al. (2021), authors show that by masking out the gradients that have opposite signs in different domains, they are able to trade-off some “learning speed” for prioritizing learning the invariances. Our principle for proposing the IDGM objective is identical, except in this case we actively *encourage* these gradients to be of consistent signs. This unfortunately, does result in computing the costly second-order derivative, which makes it difficult for us to empirically evaluate the IDGM objective on non-toy dataset². We believe that the large performance gain of IDGM over ERM on the linear and CDSPRITES datasets, along with the detailed discussion on the relationship between gradient sign consistency and invariant representation, is enough to justify for IDGM’s ability to improve domain generalization.

²For context, training an IDGM model on VLCS dataset with batch size 32 and standard RESNET18 backbone does not fit a NVIDIA V100 (32GB) GPU.

5.5 Conclusion

In this paper we presented inter-domain gradient matching (IDGM) for domain generalization. To avoid costly second-order computations, we approximated IDGM with a simple first-order algorithm, Fish. We demonstrated our algorithm’s capability to learn from invariant features (as well as ERM’s failure to do so) using simple datasets such as CDSPRITES-N and the linear example. We then evaluated the model’s performance on WILDS and DOMAINBED, demonstrating that Fish performs well on different subgenres of domain generalization, and surpasses baseline performance on a diverse range of vision and language tasks using different architectures such as DenseNet, ResNet-50 and BERT. Our experiments can be replicated with 1500 GPU hours on NVIDIA V100.

Despite its strong performance, similar to previous work on domain generalization, when the number of domains is large Fish struggles to outperform ERM. We are currently investigating approaches by which Fish can be made to scale to datasets with orders of magnitude more domains and expect to report on this improvement in our future work.

6

Adversarial Masking for Self-Supervised Learning

Abstract

We propose ADIOS, a masked image modeling (MIM) framework for self-supervised learning, which simultaneously learns a masking function and an image encoder using an adversarial objective. The image encoder is trained to minimise the distance between representations of the original and that of a masked image. The masking function, conversely, aims at maximising this distance. ADIOS consistently improves on state-of-the-art self-supervised learning (SSL) methods on a variety of tasks and datasets—including classification on ImageNet100 and STL10, transfer learning on CIFAR10/100, Flowers102 and iNaturalist, as well as robustness evaluated on the backgrounds challenge (Xiao et al., 2021)—while generating semantically meaningful masks. Unlike modern MIM models such as MAE, BEiT and iBOT, ADIOS does not rely on the image-patch tokenisation construction of Vision Transformers, and can be implemented with convolutional backbones. We further demonstrate that the masks learned by ADIOS are more effective in improving representation learning of SSL methods than masking schemes used in popular MIM models. Code is available at <https://github.com/YugeTen/adios>.

6.1 Introduction

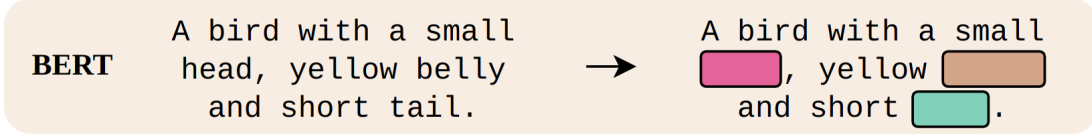
The goal of Masked image modeling (MIM) is to learn image representations, in a self-supervised fashion, by occluding parts of the input images. MIM is inspired by significant advances in natural language modelling such as BERT (Devlin et al., 2019), where the model is trained to fill-in words randomly removed from a sentence (Figure 6.1, top row). Recent work, including MAE (He et al., 2021) and BEiT (Bao et al., 2021), show that these gains are at least partially transferable to vision. The task of a MIM model is therefore similar to BERT, e.g., given an image of a bird in Figure 6.1, it needs to reason about what the bird might be sitting on or what colour the bird’s belly is given the visible context (bottom row). However, while missing words describe *whole* semantic entities (e.g. “head”), the masks used for context encoder (Pathak et al. (2016), which pioneered MIM), BEiT and MAE typically have no such constraint (Figure 6.1 bottom, left to right). Imputation under such schemes is conceptually simpler, as random masking only *partially* obscures meaningful visual entities, which allows easier inference of missing values by leveraging strong correlations at the local-pixel level¹.

To narrow the gap between pixel masking and word masking, we posit that one needs to occlude *whole entities in the image*. This encourages the model to perform imputation by complex semantic reasoning using the unmasked context (e.g. given a bird with a yellow body, it is likely to have a yellow head) rather than leveraging simple local correlations, which can benefit representation learning. Interestingly, He et al. (2021) are motivated by similar hypothesis and propose to occlude a large fraction (up to 75%) of the image, removing complete entities by a higher chance, which they find is essential for good performance. Here, we suggest that it is actually *what is masked*, not so much *how much is masked*, that is crucial for effective self-supervised representation learning.

To this end, we investigate learning to mask with an adversarial objective, where an occlusion model is asked to make reasoning about missing parts of the scene

¹Akin to randomly masking letters in a sentence for NLP.

Masked Language Model



Masked Image Models



Figure 6.1: Self-supervised language, and vision, models learn representations by imputing data removed by masking. ***BERT***: random word masks; ***Context encoder***: random, fix-shaped mask; ***BEiT***: random ‘blockwise’ masking; ***MAE***: randomly mask out 75% of the image; ***ADIOS***: multiple masks (N=3) generated by an adversarially trained masking model, post-processed with fully connected conditional random fields (Krähenbühl and Koltun, 2011).

more difficult. This novel representation-learning algorithm, called **Adversarial Inference-Occlusion Self-supervision (ADIOS)**, can identify and mask out regions of correlated pixels within an image (Figure 6.1, bottom right), which brings it closer to the word-masking regime in natural language. And as we shall see in Section 6.3, it consistently improves performance of state-of-the-art self-supervised learning (SSL) algorithms.

Some MIM methods employ a generative component for representation learning, by learning to reconstruct the masked image. However, it has been shown Bao et al. (2021); Ramesh et al. (2021) that pixel-level reconstruction tasks waste modelling capacity on high-frequency details over low-frequency structure, leading to subpar performance. We hence frame ADIOS as an encoder-only framework that minimises the distance between the *representations* of the original image and the masked image. The occlusion model, which is trained adversarially to the encoder, tries to minimise this same distance. We further discuss in Section 6.2.1 that, compared to the generative setup, the encoder-only setup optimises a functionally superior objective for representation learning. Note that the encoder objective is compatible

with many recent augmentation-based Siamese self-supervised learning (SSL; Chen et al. (2020a); Chen and He (2021)) methods. We show that ADIOS consistently improves performance of these SSL objectives, showcasing the generality of our approach.

Our main contributions are as follows,

1. A novel adversarial Siamese-style MIM framework, that unlike other MIM methods is not limited to using ViT as backbone—advantageous given recent discoveries of modernised-convnet superiority over ViTs (Liu et al., 2022b; Touvron et al., 2021);
2. Qualitative and quantitative analyses showing that masks generated by ADIOS are semantically meaningful;
3. Analysis of how different masking schemes affect representation-learning performance of SSL models. We find models trained with ADIOS and ground-truth object masks significantly outperform other masking schemes/no mask, demonstrating the efficacy of semantically meaningful masks for representation learning.

6.2 Methodology

Set up ADIOS consists of two components, inference model $\mathcal{I}$ and occlusion model $\mathcal{M}$ (see Figure 6.2). Given an RGB image $\mathbf{x}$, the occlusion model produces an image-sized mask $\mathbf{m} = \mathcal{M}(\mathbf{x})$ with values in $[0, 1]$. The inference model $\mathcal{I}$ takes original image $\mathbf{x}$ and an occluded image $\mathbf{x}^m = \mathbf{x} \odot \mathbf{m}$ ($\odot$ is the Hadamard product) as inputs, generating representations for both, which we denote as $\mathbf{z}$ and $\mathbf{z}^m$. The two models are learnt by solving for

$$\mathcal{I}^*, \mathcal{M}^* = \arg \min_{\mathcal{I}} \max_{\mathcal{M}} \mathcal{L}(\mathbf{x}; \mathcal{I}, \mathcal{M}) . \quad (6.1)$$

We will now discuss different choices for $\mathcal{I}$ and $\mathcal{M}$.

6.2.1 Inference model $\mathcal{I}$

As discussed in Section 6.1, the inference model should minimise some distance between the original and masked images. Here, we discuss potential forms of this objective, arriving at our final framework using augmentation-based SSL methods.

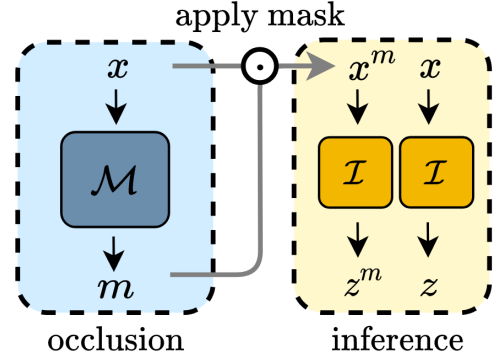


Figure 6.2: ADIOS Architecture.

Distance in pixel space One option would be to inpaint the masked image with the inference model, and train $\mathcal{I}$ by minimising the distance between the inpainted image and the original image in pixel space. More specifically, we can define $\mathcal{I}$ as an auto-encoder consisting of an encoder and decoder, which takes the masked image $\mathbf{x}^m$ as input and produces inpainted image $\hat{\mathbf{x}}$ (see Figure 6.3). The model can be trained using the following reconstruction loss

$$\mathcal{L}_{\text{AE}}(\mathbf{x}; \mathcal{I}, \mathcal{M}) = \mathcal{D}(\mathbf{x}, \hat{\mathbf{x}}) = \mathcal{D}(\mathbf{x}, \mathcal{I}(\mathbf{x} \odot \mathcal{M}(\mathbf{x}))), \quad (6.2)$$

where $\mathcal{D}$ denotes some distance metric defined in pixel space. Minimising eq. (6.2) encourages the auto-encoder to impute the missing part of the image as accurately as possible. $\mathcal{L}_{\text{AE}}$ can then be used in eq. (6.1) to train the inference-occlusion model.

Distance in representation space An interesting question for auto-encoding $\mathcal{I}$ is: where does the imputation happen? Multiple hypotheses exist: ① **Encoder only:** The encoder q completely recovers the missing information in the masked image, and in the ideal case, $q(\mathcal{M}(\mathbf{x})) = q(\mathbf{x})$; the decoder faithfully reconstructs these representations. ② **Decoder only:** The encoder faithfully extracts all information from the masked image, and the decoder reasons to inpaint missing pixel from the representations; ③ **Both:** The encoder and decoder both inpaint parts of the image.

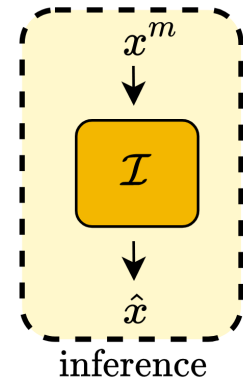


Figure 6.3: Inpainting.

Given these scenarios, ① is clearly best suited for representation learning, as it requires the encoder to reason about the missing parts based on observed context,

beyond just extracting image features. With representation learning, rather than inpainting, being our end goal, the key challenge lies in designing an objective targetting scenario ①, such that we learn the most expressive version of encoder q .

A key feature in ① is that when the encoder recovers all information of the original image, $q(\mathbf{x}^m) = q(\mathbf{x})$, the features extracted from the partially observed image $\mathbf{x}^m$ should in principle be the same as those extracted from the unmasked image $\mathbf{x}$. We thus propose an inference model $\mathcal{I}$ consisting of only the encoder, which extracts representation $\mathbf{z} = \mathcal{I}(\mathbf{x})$, $\mathbf{z} \in \mathbb{R}^d$. Our objective can thus be written as

$$\mathcal{L}_{\text{ENC}}(\mathbf{x}; \mathcal{I}, \mathcal{M}) = \mathcal{D}(\mathbf{z}, \mathbf{z}^m) = \mathcal{D}(\mathcal{I}(\mathbf{x}), \mathcal{I}(\mathbf{x} \odot \mathcal{M}(\mathbf{x}))) \quad (6.3)$$

where $\mathcal{D}$ is some distance metric defined in $\mathbb{R}^d$. Not only does $\mathcal{L}_{\text{ENC}}$ encourages the learning of more expressive encoder that can infer missing information, optimising this objective also does not involve generative component p , which is redundant for representation learning.

SSL framework eq. (6.3) can be realised by simply optimising a Siamese network (Bromley et al., 1993). However, the objective can be trivially minimised when the representations for all inputs “collapse” to a constant. This

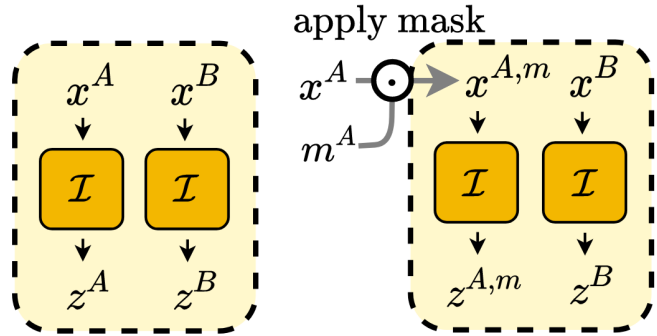


Figure 6.4: *Left:* SimCLR. *Right:* SimCLR + ADIOS.

phenomenon, known as latent collapse, has been addressed in many ways in augmentation-based SSL.

Let us take SimCLR (Chen et al., 2020a) as an example (see Figure 6.4, left); given a minibatch of M input images $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^M$, two sets of random augmentations A and B are applied to each image in $\mathbf{x}$, yielding $\mathbf{x}^A$ and $\mathbf{x}^B$. The same encoding function $\mathcal{I}$ is used to extract representations from both sets of augmented views,

yielding $\mathbf{z}^A = \mathcal{I}(\mathbf{x}^A)$ and $\mathbf{z}^B = \mathcal{I}(\mathbf{x}^B)$. The objective of SimCLR is defined as

$$\mathcal{L}_{\text{SimCLR}}(\mathbf{x}; \mathcal{I}) = \log \frac{\exp(\mathcal{D}(\mathbf{z}_i^A, \mathbf{z}_i^B))}{\sum_{i \neq j} \exp(\mathcal{D}(\mathbf{z}_i^A, \mathbf{z}_j^B))}, \quad (6.4)$$

where $\mathcal{D}$ denotes the negative cosine similarity². Intuitively, the objective minimises the distance between representations of the two augmented views of the same image (i.e. $\mathbf{z}_i^A, \mathbf{z}_i^B$), while repulsing the representations of different images (i.e. $\mathbf{z}_i^A, \mathbf{z}_j^B$). This effectively prevents the representations of different images from collapsing to the same constant, while optimising an objective similar to Equation (6.3).

We can use the SimCLR objective for our model by masking one of the augmented images and then follow the exact same pipeline (see Figure 6.4, right). More specifically, we replace $\mathbf{z}^A$ by $\mathbf{z}^{A,m} = \mathcal{I}(\mathbf{x}^A \odot \mathbf{m}^A)$, where $\mathbf{m}^A = \mathcal{M}(\mathbf{x}^A)$ is a mask generated by the occlusion model given $\mathbf{x}^A$. Following Equation (6.4), we can write the SimCLR-ADIOS objective as

$$\begin{aligned} \mathcal{L}_{\text{SimCLR}}^{\text{ADIOS}}(\mathbf{x}; \mathcal{I}, \mathcal{M}) &= \log \frac{\exp(\mathcal{D}(\mathbf{z}_i^{A,m}, \mathbf{z}_i^B))}{\sum_{i \neq j} \exp(\mathcal{D}(\mathbf{z}_i^{A,m}, \mathbf{z}_j^B))} \\ &= \log \frac{\exp(\mathcal{D}(\mathcal{I}(\mathbf{x}_i^A \odot \mathcal{M}(\mathbf{x}_i^A)), \mathcal{I}(\mathbf{x}_i^B)))}{\sum_{i \neq j} \exp(\mathcal{D}(\mathcal{I}(\mathbf{x}_i^A \odot \mathcal{M}(\mathbf{x}_i^A)), \mathcal{I}(\mathbf{x}_j^B)))}. \end{aligned} \quad (6.5)$$

Again, we can use eq. (6.5) in Equation (6.1) to train the inference-occlusion model. Crucially, any SSL method that compares two augmented image views includes a term like Equation (6.3), and can be plugged in to our framework. We conduct experiments using SimCLR, BYOL Grill et al. (2020), and SimSiam Chen and He (2021) objectives and show significant improvement on downstream task performance with each method. Refer to Appendix E.1 for the ADIOS objective used for BYOL and SimSiam, as well as more details on the SimCLR objective.

6.2.2 Occlusion model $\mathcal{M}$

For simplicity, we only consider the single-mask-generating case in the discussion above. In practice, since an image typically contains multiple components, we generate $N > 1$ masks to challenge the model to reason about relations between different components—empirical performance confirm benefits of doing so.

²We omit the SimCLR temperature parameter for simplicity.

There are many parametric forms $\mathcal{M}$ could employ. For instance, one could consider generating multiple masks sequentially in an auto-regressive manner as seen in Engelcke et al. (2021). However, we find that the simplest setup suffices, where $\mathcal{M} : \mathbb{R}^{c \times w \times h} \mapsto \mathbb{R}^{N \times w \times h}$ consists of a learnable neural network and a pixelwise softmax layer σ applied across N masks to ensure that the sum of a given pixel across all the masks equals 1. We use U-Net as the backbone of our occlusion model—see Appendix E.2 for more details. Note that we experimented with binarising the masks during training, but found that this did not yield improvements, and hence used real-valued masks directly.

6.2.3 Putting it together

Here we present ADIOS in its complete form. This includes the N -mask occlusion model, which generates masks $\{\mathbf{m}^{(n)}\}_{n=1}^N$ from the RGB image $\mathbf{x}$. The inference model computes a loss $\mathcal{L}^{(n)}(\mathbf{x}; \mathcal{I}, \mathcal{M})$ for each $\mathbf{m}^{(n)}$ and

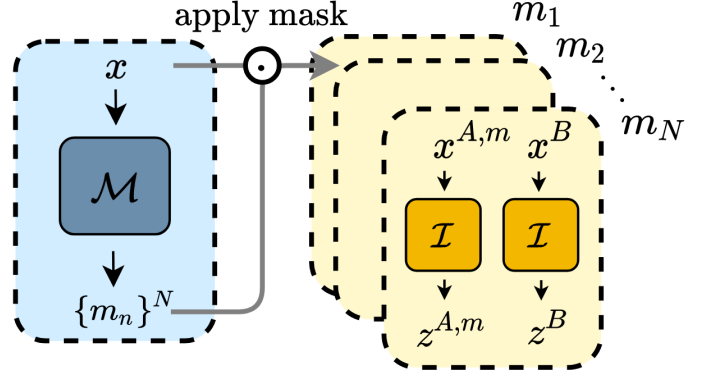


Figure 6.5: ADIOS, $N > 1$.

the final loss is computed by averaging across N masks

$$\mathcal{I}^*, \mathcal{M}^* = \arg \min_{\mathcal{I}} \max_{\mathcal{M}} \frac{1}{N} \sum_{n=1}^N \mathcal{L}^{(n)}(\mathbf{x}; \mathcal{I}, \mathcal{M}). \quad (6.6)$$

We provide PyTorch-style pseudo code for the training of ADIOS in Algorithm 3.

Sparsity penalty A trivial solution exists for this objective Equation (6.6), where, for masks $\{\mathbf{m}^{(1)}, \dots, \mathbf{m}^{(N)}\}$, some mask $\mathbf{m}^{(n)}$ occludes everything, with the other $\{N\}_{\setminus n}$ masks not occluding anything. To avoid such degenerate solutions, we introduce a sparsity penalty p_n in the form of $1/\sin(\cdot)$ that discourages the occlusion model from generating all-one or all-zero masks; specifically,

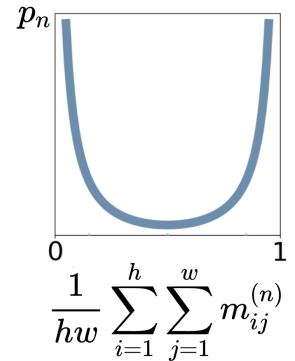


Figure 6.6: Penalty. (6.7)

$$p_n = \sin \left(\frac{\pi}{hw} \sum_{i=1}^h \sum_{j=1}^w m_{ij}^{(n)} \right)^{-1}.$$

Algorithm 3: Pseudo code of ADIOS.

```

# Iterate over dataloader
for x in dataloader:
    # Perform data augmentation
    x, x_aug = augment_image(x)
    # Generate masks
    masks = softmax(occlusion_model(x))
    # Initialize loss and gradients
    loss = 0.0
    optimizer.zero_grad()
    # Iterate over masks
    for mask in masks:
        # Apply mask to input
        x_masked = x * mask
        # Encode masked input and augmented input
        z_masked = inference_model(x_masked)
        z_aug = inference_model(x_aug)
        # Compute loss
        loss += objective(z_masked, z_aug)
    # Backpropagation and optimization step
    loss.backward()
    optimizer.step()

```

Note that, p_n goes to infinity as $\mathbf{m}^{(n)}$ approaches all-one or all-zero (see Figure 6.6). Minimising p_n with respect to $\mathcal{M}$ encourages the occlusion model to generate semantically meaningful mask, while avoiding degenerate solutions.

Final objective Let λ the scaling of the penalty term. Our complete objective reads as

$$\mathcal{I}^*, \mathcal{M}^* = \arg \min_{\mathcal{I}} \max_{\mathcal{M}} \frac{1}{N} \sum_{n=1}^N \left(\mathcal{L}^{(n)}(\mathbf{x}; \mathcal{I}, \mathcal{M}) - \lambda p_n \right). \quad (6.8)$$

Lightweight ADIOS Despite its strong empirical performance, we note that the training objective in Equation (6.8) requires N forward passes, which can be computationally expensive as we increase N for more complex data. We therefore develop a lightweight version of ADIOS, where we randomly sample one from the N generated masks to be applied to the input image. Doing so disassociates the computational cost of the model from the number of generated masks, and the only cost increase comes from applying the mask generation model once, which is

inexpensive (10% the size of ResNet18). We name this single-forward pass version of our model ADIOS-s, and write the objective as

$$\mathcal{I}^*, \mathcal{M}^* = \arg \min_{\mathcal{I}} \max_{\mathcal{M}} \left(\mathcal{L}^{(k)}(\mathbf{x}; \mathcal{I}, \mathcal{M}) - \lambda \frac{1}{N} \sum_{n=1}^N p_n \right),$$

(6.9)

where $k \sim \text{Uniform}(\{1, 2, \dots, N\})$.

6.3 Evaluation of Representations

Set up We evaluate ADIOS with three different SSL objectives: SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), and SimSiam (Chen and He, 2021). Each set of quantitative results is reported as an average over three random trials. We summarise our training setup in Appendix E.3.

6.3.1 Classification

We evaluate the performance of ADIOS on STL10, as well as a downsized version of ImageNet100 (Tian et al., 2020b), from resolution 224x224 to 96x96. We refer to our version of the dataset as ImageNet100-S. We also evaluate the performance of ADIOS-s on the original ImageNet100 dataset. Both ImageNet100 and STL10 are derived from ImageNet-1k (Deng et al., 2009): ImageNet100 contains data from 100 ImageNet classes, and STL10 is derived from 10 object classes of ImageNet, with 5,000 labelled images and 100,000 unlabelled images. Due to computational constraints we were unable to evaluate on ImageNet-1k; we leave this for future work.

For ADIOS, we provide results using ResNet18 (He et al., 2016) and ViT-Tiny (Dosovitskiy et al., 2021) backbones on three classification tasks: linear evaluation, k -NN and clustering. Through hyperparameter search, we use $N = 4$ masking slots for ImageNet100-S and $N = 6$ for STL10. For ADIOS-s we provide results using ResNet18 as backbone on linear evaluation.

Linear evaluation and k -NN We study the utility of learned representation by classifying the features using both linear and a k -nearest neighbour (k -NN) classifiers. Following the protocol in Zhou et al. (2021), we sweep over different

Table 6.1: Top-1 classification accuracy (k -NN and Linear Probing) on Imagenet100-S, STL10. Improvements of ADIOS that are more than 3% are marked in bold.

Method	ImageNet100-S		STL10	
	k -NN	Linear	k -NN	Linear
<i>Backbone: ViT-Tiny</i>				
SimCLR	40.0 (± 0.28)	40.2 (± 0.47)	72.9 (± 0.27)	76.0 (± 0.33)
+ADIOS	42.0 (± 1.32)	43.1 (± 0.71)	73.4 (± 0.28)	79.7 (± 0.88)
SimSiam	35.2 (± 1.12)	36.8 (± 1.82)	66.7 (± 0.10)	67.5 (± 0.02)
+ADIOS	38.8 (± 2.73)	40.1 (± 0.59)	67.9 (± 0.75)	68.8 (± 0.25)
BYOL	38.1 (± 0.61)	39.7 (± 0.50)	71.9 (± 0.12)	72.1 (± 0.32)
+ADIOS	47.1 (± 0.35)	49.2 (± 0.94)	74.5 (± 0.58)	75.9 (± 0.63)
<i>Backbone: ResNet-18</i>				
SimCLR	54.1 (± 0.09)	55.1 (± 0.15)	83.7 (± 0.48)	85.1 (± 0.12)
+ADIOS	55.1 (± 0.43)	55.9 (± 0.21)	85.8 (± 0.10)	86.1 (± 0.36)
SimSiam	58.6 (± 0.31)	59.5 (± 0.31)	84.3 (± 0.81)	84.8 (± 0.72)
+ADIOS	61.0 (± 0.29)	60.4 (± 0.19)	84.6 (± 0.35)	86.4 (± 0.24)
BYOL	56.2 (± 0.79)	56.3 (± 0.10)	83.6 (± 0.09)	84.3 (± 0.13)
+ADIOS	60.2 (± 0.82)	61.4 (± 0.14)	84.8 (± 0.19)	85.6 (± 0.24)

numbers of nearest neighbours for k -NN and different learning rates for the linear classifier.

Results are presented in Table 6.1, where each *+ADIOS* entry represents the ADIOS framework applied to the SSL objective in the row above. For instance, the top coloured block shows results of SimCLR and SimCLR+ADIOS. Models using ADIOS consistently outperform their respective SSL baselines beyond the margin of error; in some cases achieving significant improvements of 3–9% (in bold). Notably, the ViT-Tiny models perform significantly worse than ResNet-18 models, which is unsurprising given that ViT-Tiny uses half the number of parameters of ResNet-18.

The best Top-1 accuracy on ImageNet100-S is 61.4%, achieved by BYOL+ADIOS using linear evaluation, surpassing its baseline BYOL by 5.1%; For STL10, the best performing model is SimSiam+ADIOS using linear evaluation with an accuracy of 86.4%, while SimSiam evaluates at 84.8%. Significantly, ADIOS improves all metrics on both backbones and both datasets. Interestingly, the degree of improvement varies by method, and can result in order change between respective SSL methods.

We also run experiments on the original ImageNet100 dataset with the single-forward pass version of our model, ADIOS-s. The results in Table 6.2 show that this much

Table 6.2: Top-1 classification accuracy of linear probing on ImageNet100. Improvements of more than 3% are marked in bold.

SimCLR	+ADIOS-s	SimSiam	+ADIOS-s	BYOL	+ADIOS-s
77.5 (± 0.10)	76.1 (± 0.50)	76.4 (± 0.07)	77.2 (± 0.09)	74.3 (± 0.16)	80.8 (± 0.60)

cheaper model also achieves impressive performance, especially when applied to BYOL with a performance boost of more than 6%. This result further demonstrates the efficiency of our approach, and the reduced computational cost allows for the potential of scaling to larger datasets.

Clustering Following Bao et al. (2021); Zhou et al. (2021), we also evaluate the trained models using standard clustering metrics, including adjusted random index (ARI) and Fowlkes-Mallows index (FMI), both of which computes the similarity between clusterings, as well as normalised mutual information (NMI). We assign pseudo-labels to the representation of each image using k-means, and evaluate the three metrics on the clusters formed by the pseudo-labels vs. the true labels. Results in Table 6.3 are consistent with our previous findings. ADIOS improves the performance of baseline SSL methods on all three metrics for both datasets.

Findings ADIOS significantly and consistently improves the quality of representation learned under a range of set-ups, across two datasets, two backbone architectures, three SSL methods and on five different metrics, highlighting the effectiveness and versatility of our approach.

6.3.2 Transfer learning

We study the downstream performance of models trained on ImageNet100-S, on four different datasets including CIFAR10, CIFAR100 (Krizhevsky et al., 2009), Flowers102 (Nilsback and Zisserman, 2008), and iNaturalist (Horn et al., 2018). CIFAR10 and CIFAR100 resolutions are 32, while those of Flowers102 and iNaturalist are 96. We only use the ResNet-18 models here as they clearly outperform the ViT-Tiny models in our previous experiments. Detailed setup and hyperparameters are given in Appendix E.4.

Table 6.3: Clustering performance on Imagenet100-S, STL10.

Method	Backbone	Metrics		
		FMI $\uparrow$	ARI $\uparrow$	NMI $\uparrow$
<i>Dataset: ImageNet100-S</i>				
SimCLR	ViT-Tiny	0.105 ($\pm 1e-3$)	0.095 ($\pm 1e-3$)	0.432 ($\pm 3e-3$)
+ADIOS	ViT-Tiny	0.120 ($\pm 1e-3$)	0.110 ($\pm 1e-3$)	0.442 ($\pm 4e-3$)
SimSiam	ViT-Tiny	0.077 ($\pm 9e-4$)	0.067 ($\pm 2e-3$)	0.389 ($\pm 3e-3$)
+ADIOS	ViT-Tiny	0.098 ($\pm 1e-2$)	0.087 ($\pm 9e-4$)	0.425 ($\pm 3e-3$)
BYOL	ViT-Tiny	0.098 ($\pm 8e-3$)	0.088 ($\pm 8e-3$)	0.418 ($\pm 4e-3$)
+ADIOS	ViT-Tiny	0.132 ($\pm 3e-3$)	0.123 ($\pm 1e-3$)	0.458 ($\pm 4e-3$)
SimCLR	ResNet18	0.151 ($\pm 3e-3$)	0.135 ($\pm 4e-3$)	0.515 ($\pm 6e-3$)
+ADIOS	ResNet18	0.175 ($\pm 1e-3$)	0.161 ($\pm 4e-3$)	0.539 ($\pm 3e-3$)
SimSiam	ResNet18	0.167 ($\pm 2e-3$)	0.136 ($\pm 6e-3$)	0.553 ($\pm 8e-3$)
+ADIOS	ResNet18	0.179 ($\pm 1e-3$)	0.161 ($\pm 1e-3$)	0.553 ($\pm 1e-3$)
BYOL	ResNet18	0.170 ($\pm 1e-3$)	0.158 ($\pm 3e-3$)	0.530 ($\pm 4e-3$)
+ADIOS	ResNet18	0.179 ($\pm 6e-4$)	0.156 ($\pm 2e-3$)	0.561 ($\pm 2e-3$)
<i>Dataset: STL10</i>				
SimCLR	ViT-Tiny	0.349 ($\pm 5e-3$)	0.269 ($\pm 6e-3$)	0.410 ($\pm 2e-3$)
+ADIOS	ViT-Tiny	0.351 ($\pm 4e-3$)	0.271 ($\pm 8e-3$)	0.417 ($\pm 6e-3$)
SimSiam	ViT-Tiny	0.296 ($\pm 3e-3$)	0.177 ($\pm 1e-3$)	0.341 ($\pm 4e-3$)
+ADIOS	ViT-Tiny	0.320 ($\pm 3e-3$)	0.235 ($\pm 5e-3$)	0.349 ($\pm 0e-0$)
BYOL	ViT-Tiny	0.349 ($\pm 5e-3$)	0.269 ($\pm 5e-3$)	0.410 ($\pm 5e-3$)
+ADIOS	ViT-Tiny	0.355 ($\pm 4e-2$)	0.276 ($\pm 3e-3$)	0.422 ($\pm 4e-3$)
SimCLR	ResNet18	0.338 ($\pm 2e-3$)	0.166 ($\pm 9e-4$)	0.512 ($\pm 5e-3$)
+ADIOS	ResNet18	0.437 ($\pm 6e-3$)	0.309 ($\pm 9e-3$)	0.585 ($\pm 8e-3$)
SimSiam	ResNet18	0.392 ($\pm 2e-3$)	0.242 ($\pm 7e-3$)	0.552 ($\pm 3e-3$)
+ADIOS	ResNet18	0.412 ($\pm 8e-3$)	0.249 ($\pm 7e-3$)	0.558 ($\pm 2e-4$)
BYOL	ResNet18	0.429 ($\pm 5e-3$)	0.328 ($\pm 9e-3$)	0.525 ($\pm 8e-3$)
+ADIOS	ResNet18	0.508 ($\pm 6e-3$)	0.422 ($\pm 1e-2$)	0.588 ($\pm 9e-3$)

Table 6.4 reports classification accuracy under two different transfer-learning setups, including *F.T.*: fine-tuning the entire model and *Lin.*: freeze the encoder weights and re-train the linear classifier only. As a comparison, we also show the results of training from scratch on each dataset.

Results show that ADIOS improves transfer learning performance on all four datasets, under both linear evaluation and fine-tuning. Bigger improvements, of $>3\%$ (marked in bold) occur mostly under linear evaluation, indicating that compared to baseline SSL models, the ADIOS-pre-trained representations are much easier to linearly separate. Notably, all 6 models’ fine-tuning performance exceeds training from scratch, demonstrating the benefits of pretraining.

One might also notice that different from the other datasets, there exists large discrepancies between the linear evaluation vs. fine-tuning performance on CIFAR10

Table 6.4: Classification accuracy of transfer learning by re-training linear classifier only (*Lin.*) and fine-tuning (*F.T.*). More than 3% improvements by ADIOS are marked in bold.

Method	CIFAR10		CIFAR100		Flowers102		iNaturalist	
	Lin.	F.T.	Lin.	F.T.	Lin.	F.T.	Lin.	F.T.
SimCLR	30.1	91.3	10.2	70.0	42.5	45.6	69.4	82.1
+ADIOS	34.6	93.4	11.0	71.8	50.2	50.6	72.5	84.3
SimSiam	35.3	92.4	13.2	65.0	38.7	55.0	72.3	85.0
+ADIOS	39.3	94.3	13.3	71.0	44.9	59.0	75.9	86.2
BYOL	29.9	88.0	13.3	52.3	49.1	58.6	72.7	85.1
+ADIOS	39.2	90.4	14.0	62.0	51.7	60.1	73.1	85.7
Scratch	-	85.5	-	49.8	-	30.6	-	73.8

and CIFAR100. This is because He et al. (2016) suggests a slightly different architecture for CIFAR with smaller kernel size in the first layer to suit its small image size (see details in Appendix E.6). We use CIFAR-ResNet for fine-tuning, however we have to use the original architecture for linear evaluation in order to use the pretrained weights, which leads to poor performance.

6.3.3 Robustness

It is likely that the adversarial masks learned by ADIOS target informative image features, including spurious correlations if present in a dataset. It is therefore interesting to ask if ADIOS representations are robust to changes in such spurious correlations. To answer this, we evaluate models pretrained on ImageNet100-S on the backgrounds challenge (Xiao et al., 2021), where 7 different types of variation on a subset of ImageNet data are used to measure the impact of

foreground and background on model decision making. Examples of such variations can be seen in Figure 6.7, where the original figure’s (*Orig.*) background is replaced by background from another image in the same class (*M.S.*), from a random image in any class (*M.R.*) or from an image in the next class (*N.R.*). We perform linear evaluation on the pretrained models on these variations, and report the classification accuracy in Table 6.5.

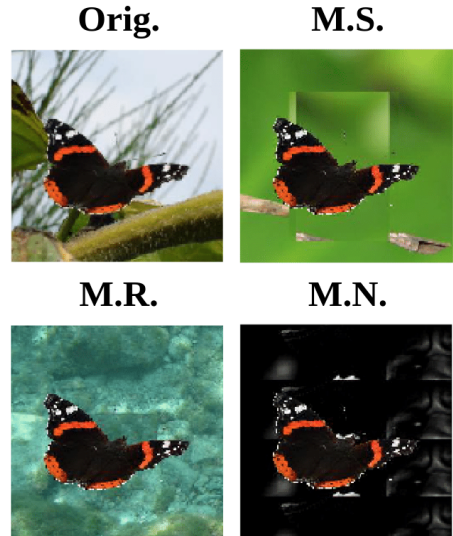
**Figure 6.7:** Examples.

Table 6.5: Accuracy on different variations of the backgrounds challenge, evaluating model robustness. Example variations in Figure 6.7.

Method	Variations							Orig.
	O.BB.	O.BT.	N.F.	O.F.	M.S.	M.R.	M.N.	
SimCLR	20.1	34.8	44.3	41.6	67.1	45.9	41.0	78.8
+ADIOS	20.7	36.7	45.5	43.5	68.0	47.9	43.7	79.1
SimSiam	29.5	39.1	43.8	52.1	69.9	43.9	40.8	78.4
+ADIOS	33.1	41.0	46.2	54.7	71.5	47.2	43.5	80.3
BYOL	25.9	38.4	46.0	51.6	71.3	45.6	42.7	79.8
+ADIOS	27.7	39.0	48.5	51.7	72.1	47.8	44.1	80.6
Avg. gain	+2.0	+1.4	+2.0	+1.5	+1.1	+2.5	+2.3	+1.0

Our results show that all three SSL-ADIOS models outperform their respective baselines on *all* variations, demonstrating that ADIOS-learned representations are more robust to changes in both foreground and background.

It is useful to examine how ADIOS behaves in different testing conditions regardless of the SSL objective used. To this end, the bottom row of Table 6.5 contains the performance gain of ADIOS averaged over all three SSL models. We witness the biggest gains on the *M.R.* and *M.N.* conditions (bottom row, Figure 6.7). That is, when any deterministic relation between labels and backgrounds is severed. The improvement is the lowest for the original images and *M.S.* condition (top row, Figure 6.7), both of which preserve the relation between labels and background. This demonstrates that ADIOS depends less on background information that are spuriously-correlated with object labels when making predictions. This is not surprising—as we describe in Section 6.4.1—ADIOS’ generated masks tend to occlude backgrounds, encouraging the model to focus on the foreground objects.

6.4 Analysis on Learned Masks

Here, we look at the masks generated by ADIOS’ occlusion model when trained on ImageNet100-S, STL10, and CLEVR (Johnson et al., 2017)—a dataset of rendered 3D objects such as cubes and spheres. We use $N = 4$ masking slots for CLEVR and include training details in Appendix E.5. The top row of each image block in Figure 6.8 shows the original image. The bottom row displays the generated masks, with each colour representing one masking slot. See Section 6.4.1 for a detailed analysis. We also quantitatively analyse ADIOS’ masks in Section 6.4.2.

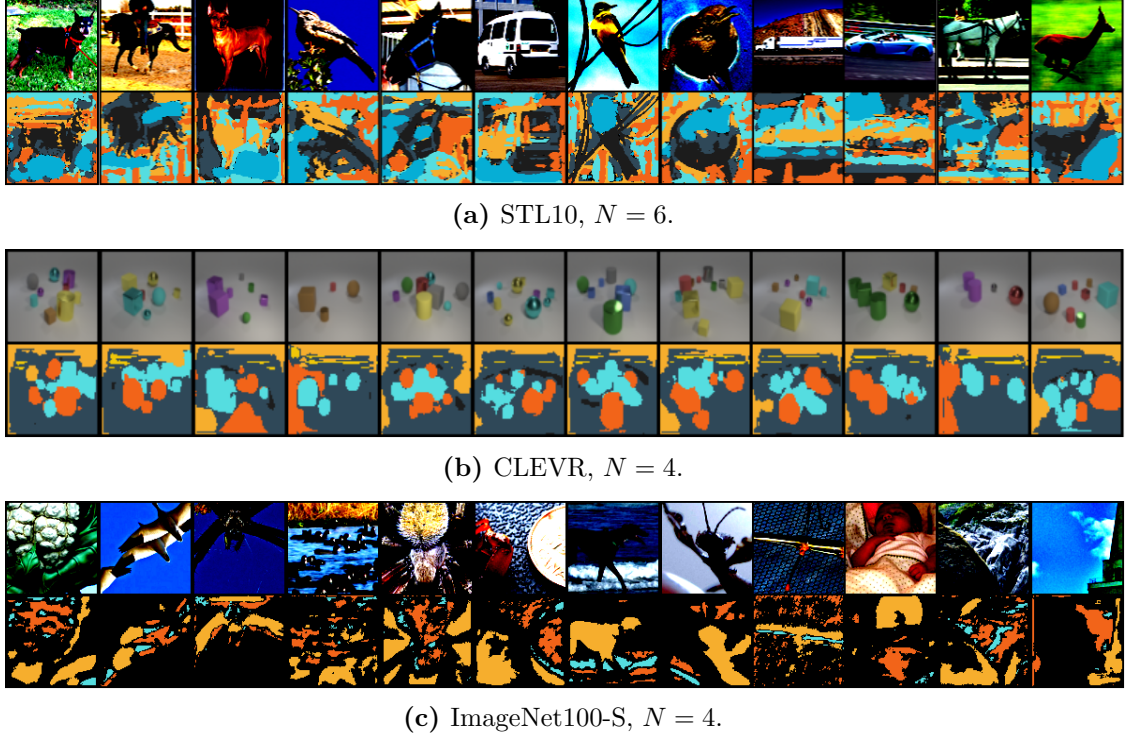


Figure 6.8: Masks generated by ADIOS during training on each dataset. N denotes the number of masks. *Top row:* original image; *Bottom row:* generated masks, each color represents one mask.

6.4.1 Mask Generation

For realistic, single-object datasets such as STL10 and ImageNet100-S, ADIOS manages to mask out different compositions of the image. In the STL10 dataset, each image clearly shows a ‘foreground’ object and a ‘background’ (Figure 6.8a). In this setting, ADIOS learns to mask specific object parts like the wings or the tail of a bird (4th column) and the mouth or the face of a horse (5th column). In case of the ImageNet100-S dataset, however, it is often not obvious if the image features any particular entity. Hence, ADIOS tends to occlude complete entities like the animal and the ant in the 7th and 8th columns of Figure 6.8c.

In CLEVR (simple rendered objects), ADIOS is usually able to put the background into a separate slot; the remaining slots split all present objects into 2-3 groups, with a tendency of applying a single mask to objects of the same colour (see Figure 6.8b). While not a focus on our work, and in contrast to prior art Greff et al. (2019);

Engelcke et al. (2020), ADIOS does not produce perfect segmentations. It is however an interesting research direction, given that better segmentations could further improve representation learning performance as we show in Section 6.4.2.

Summary Qualitative results show that ADIOS can generate semantically-meaningful masks. Crucially, the generated masks focus on different levels of detail, depending on the dataset. This may explain some of the ADIOS’ performance gains in the robustness experiments of Section 6.3.3, as semantic perturbations are baked into the training process.

6.4.2 Comparing Masking Schemes

The premise of our work is that for masked image modeling, *what is masked*, matters. In this section, we further investigate this by comparing the representation learning performance of SSL models trained under an ADIOS-like MIM framework, but with non-parametric masks including ground-truth semantic masks and random masks.

In Figure 6.9 we outline the different masking schemes investigated in this section on the CLEVR dataset, including: a) ground-truth object segmentation masks (provided with the dataset), b) foreground-background masks (where the foreground is the union of all objects), c) ground-truth, box-shaped masks, d) shuffled ground-truth object segmentation masks (i.e. one image uses the ground-truth mask of another), e) random mask occluding 75% of the image, as in MAE (He et al., 2021), f) blockwise mask occluding 30% of the image, same as BEiT (Bao et al., 2021). Note that out of these masking schemes, a–c include semantic information, while d–f do not. In addition, we perform similar experiments on ImageNet100-S and STL10, however since we do not have information of the ground-truth object segmentation, we only compare the performance of ADIOS against using the random masking scheme in MAE and BEiT (i.e. e and f).

The representation learning performance of different masking schemes on ImageNet100-S and STL10 is evaluated by its top-1 classification accuracy under linear probing. To evaluate this on CLEVR, we set up a challenging multi-label classification task.

Table 6.6: Top-1 classification accuracy on ImageNet100-S and STL10 under different masking schemes, averaged over three runs of SimCLR, SimSiam and BYOL respectively. Best results for each metric in bold.

Mask type	Condition	Dataset	
		ImageNet100-S	STL10
Random	e) MAE	43.7 (± 0.43)	78.4 (± 0.91)
	f) BEiT	46.4 (± 0.67)	80.7 (± 1.00)
Learned	ADIOS	59.2 (± 2.92)	86.0 (± 0.40)
None	-	57.0 (± 2.27)	84.7 (± 0.40)

Namely, we predict 24 binary labels, each indicating presence of a particular colour and shape (8×3) combination in the image. We report the F1 score (harmonic mean of the precision and recall) under different weighted average of subpopulation (defined by labels), where ‘micro’ evaluates F1 across the entire dataset, ‘macro’ evaluates an unweighted average of per-label F1 score, and ‘weighted’ scale the per-label F1 score by number of examples when taking average.

Tables 6.6 and 6.7 contains results averaged across these three SSL methods, each with three random trials (i.e. each entry in Tables 6.6 and 6.7 is averaged over nine runs). This is to marginalize out particular SSL methods, and therefore better understand effect of each of the masking schemes.

The results clearly show the advantage of using semantic masks over non-semantic masks for representation learning. In Table 6.6, we show that ADIOS significantly outperform random masking schemes used in MAE and BEiT; additionally, the ground-truth object masks (*G.T.*) In Table 6.7 achieves the best performance on all three metrics, closely followed by ADIOS with comparable F1-macro, F1-weighted score, and slightly lower F1-micro score. We compare this to the randomly shuffled ground-truth mask (*Shuffle*), covering on average the same image fraction as the ground-truth, but where there is far less semantic consistency to the image content. The shuffled masks perform much worse on all three metrics, supporting our hypothesis from Section 6.1 that *what is masked* is more important than *how much is masked*.

Table 6.7: Multi-label classification on CLEVR under different masking schemes, averaged over three runs of SimCLR, SimSiam and BYOL respectively. Best results for each metric in bold.

Mask type	Condition	Metric		
		F1-macro $\uparrow$	F1-micro $\uparrow$	F1-weighted $\uparrow$
Semantic	a) G.T.	0.373 ($\pm 7e-3$)	0.401 ($\pm 2e-4$)	0.460 ($\pm 1e-2$)
	b) FG./BG.	0.346 ($\pm 7e-3$)	0.365 ($\pm 2e-4$)	0.402 ($\pm 1e-3$)
	c) Box	0.347 ($\pm 2e-4$)	0.391 ($\pm 3e-5$)	0.457 ($\pm 5e-2$)
Random	d) Shuffle	0.332 ($\pm 6e-3$)	0.360 ($\pm 8e-4$)	0.418 ($\pm 1e-3$)
	e) MAE	0.309 ($\pm 8e-4$)	0.336 ($\pm 3e-4$)	0.391 ($\pm 9e-4$)
	f) BEiT	0.274 ($\pm 1e-3$)	0.307 ($\pm 2e-4$)	0.395 ($\pm 7e-3$)
Learned	ADIOS	0.377 ($\pm 2e-3$)	0.385 ($\pm 9e-4$)	0.451 ($\pm 1e-3$)
None	-	0.352 ($\pm 9e-3$)	0.359 ($\pm 2e-4$)	0.373 ($\pm 2e-5$)

The remaining masking schemes behave as expected, with the semantically-informed ones outperforming the non-semantic ones. Perhaps surprisingly, random masks, including the ones used in MAE and BEiT, can hurt representation learning under this ADIOS-like MIM framework: in both Tables 6.6 and 6.7, the performance of random masking schemes are much lower even compared to the baseline where no mask is applied. We do note that MAE and BEiT both contain many components beyond their masking schemes that are critical for their successful learning as reported in the respective works. These include image decoders, discrete-VAE tokeniser and the ViTencoder. Our evaluation focus exclusively on the masking schemes, and suggests that semantically meaningful masks lead to better representations, while random masks do not.

Summary Our experiments here show two things. Firstly, that semantically meaningful masks can be used as an effective form of augmentation for SSL models, but the same cannot be said for random masks. And secondly, that the representations learned from using the masks generated by ADIOS are comparable in quality to those learned from using ground-truth object masks.

6.5 Related Work

Augmentation based SSL Recent work has seen rapid development in SSL utilising image augmentation, with the core idea being that the representation of two augmented views of the same image should be similar. This involves a range of

work that adopts a contrastive framework where the positive sample pairs (i.e. two views of the same image) are attracted and negative pairs (i.e. two views of different images) are repulsed Chen et al. (2020a); Gansbeke et al. (2020); He et al. (2020a); Chen et al. (2020); Caron et al. (2020), as well as non-contrastive approaches Grill et al. (2020); Ermolov et al. (2021); Zbontar et al. (2021); Chen and He (2021); Bardes et al. (2021) that are able to prevent latent collapse without negative pairs, which is considered to be computationally expensive.

Learning augmentations Several models have proposed to learn augmentation policies with supervision signals Cubuk et al. (2019); Hataya et al. (2020) that is more favourable to the task at hand. Tamkin et al. (2021) applies these for SSL and proposes to learn perturbation to the input image with a *viewmaker* model that is trained adversarially to the main encoder network. Different from our work where masks are generated to occlude different components of the image, their learned perturbation is l_p -bounded and provides more “color-jitter” style augmentation to the input. Koyama et al. (2021) also proposes to learn mask-like augmentations by maximising a lower bound to the mutual information between image and representation while regularising the entropy of the augmentation. However their experiment is limited to an edited MNIST dataset, and they only consider learning augmentations for one SSL algorithm, SimCLR.

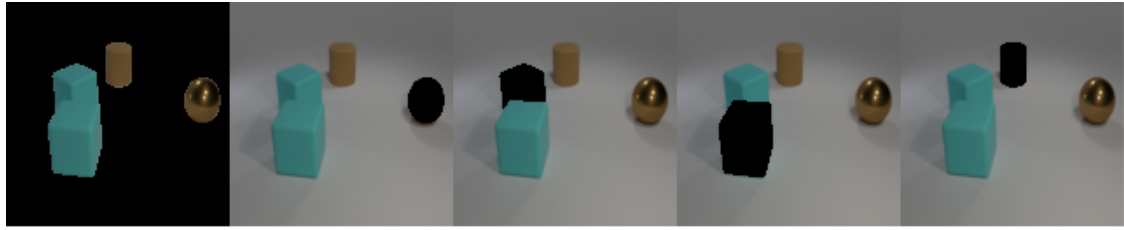
Masked image models More recently, models such as MAE (He et al., 2021), BEiT (Bao et al., 2021), iBOT (Zhou et al., 2021) have been motivated by masked language models like BERT, as we ourselves do, and achieved highly competitive results on SSL. All three methods make use of vision transformers and propose to “inpaint” images occluded by *random masks* in one way or another: MAE employs an autoencoder to inpaint images that are heavily masked, with the decoder discarded after pretraining. On the other hand, BEiT and iBOT both utilise tokenisers to first transform image patches into visual tokens, with BEiT using off-the-shelf pretrained tokenizer and iBOT training the tokeniser online. Similar to our work, rather than performing reconstruction in pixel space, they minimise the distance between the visual tokens of the complete image vs. the masked image. Recent work has also seen

the application of random masks to modalities beyond vision including speech and text, achieving strong performance in all these domains (Baevski et al.). Our work is significantly different from all above since we employ *semantically meaningful masks* from the occluder model, jointly learned with the encoding model. Moreover, our model does not rely on additional components such as tokenisers/image decoder, and since the construction of the model does not require splitting the image into patches, is also not limited to using vision transformers as the backbone architecture.

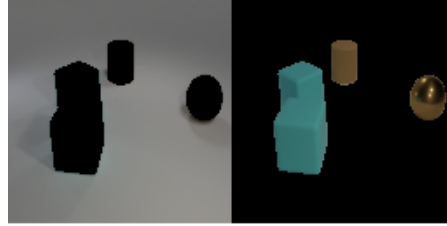
6.6 Conclusion

We propose a novel MIM framework named ADIOS, which learns a masking function alongside an image encoder in an adversarial manner. We show, in extensive experiments, that our model consistently outperforms SSL baselines on representation learning tasks, while producing semantically-meaningful masks. We also provide detailed analysis on using different forms of occlusion as augmentation for SSL in general. We find that the best representation learning performance results from using semantically-meaningful masks, especially ground-truth ones, and that masks generated by the ADIOS’ occlusion model are almost as good.

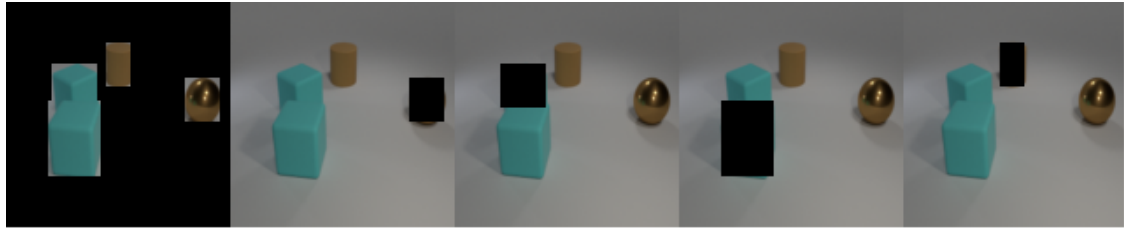
One caveat of our model is that the memory and computation cost increases linearly with the number of masking slots N , since the model requires N forward passes before each gradient update. This likely can be addressed by randomly sampling one mask for each forward pass, but is left to future work. Additionally, we want to investigate ADIOS performance on larger datasets such as ImageNet-1K or 22K and larger backbones like ViT-L, ViT-H. However, we believe that, as it is, ADIOS’ strong performance on a variety of tasks under versatile conditions provides valuable insights on the design of masked image models. Future work on masked image modeling should consider not only objectives and model architectures, but also mask design—this work shows that semantic masks are significantly more helpful than random ones in aiding representation learning.



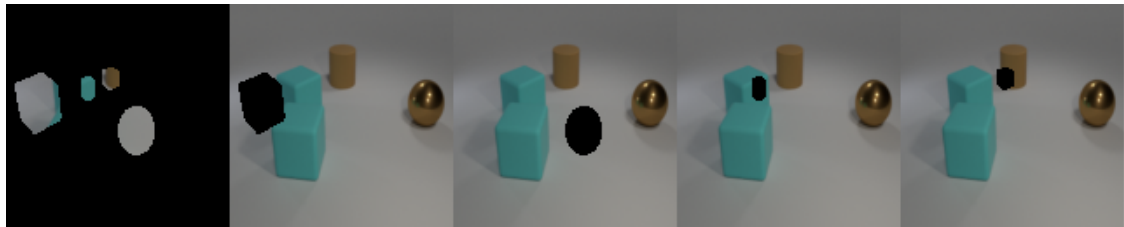
(a) Ground-truth object masks



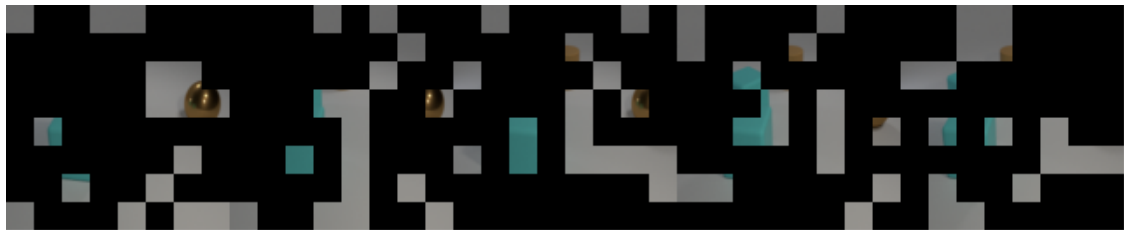
(b) Foreground-background masks



(c) Ground-truth box-shaped masks



(d) Shuffled ground-truth masks



(e) MAE (He et al., 2021) masks



(f) BEiT (Bao et al., 2021) masks

Figure 6.9: Masking schemes used to compare against ADIOS.

7

Epilogue

7.1 Summary and impact

In this thesis, we have explored the challenges and potential solutions in representation learning using multiple sources of data. We address the limitations of supervised learning, which is costly and often leads to poor generalisation and robustness, by extracting common abstract concepts between different modalities or domains to achieve more accurate and generalisable representations.

Multimodal VAEs We begin by introducing MMVAE (Chapter 3), a multimodal VAE that learns high-quality representations for diverse input modalities. This model stands out as the first multimodal VAE capable of cross-modal generation without auxiliary components. Furthermore, we propose four desiderata for multimodal VAEs, which have become widely adopted in the field (Sutter et al., 2021; Joy et al., 2020; Sutter et al., 2020). This work is widely accepted as one of the foundational work in multimodal VAE: building on our proposed objective, other works considered combining mixture and product factorisation of joint posterior (Sutter et al., 2021), as well as using a Gumbel-softmax prior (Shen et al., 2021); our mixture-of-experts factorisation for multimodal data is applied in domains such as medical imaging, sensor fusion etc. (Minoura et al., 2021; Wu and Zhuang, 2021; Zhou et al., 2022; Ye and Bors, 2021).

Contrastive multimodal learning Building upon our previous work, we also look to improve the data efficiency of such models in Chapter 4. We introduce a contrastive framework that drastically reduces the number of labeled multimodal pairs required for achieving comparable performance. This approach was innovative for multimodal learning at the time; later, with increased computing power and data availability, many more works on large-scale multimodal learning utilise contrastive loss to align the representations learned from multiple modalities. Popular among them are works such as CLIP (Radford et al., 2021), VideoCLIP (Xu et al., 2021), ALIGN (Jia et al., 2021a) and ALBEF (Li et al., 2021).

Gradient alignment for domain generalisation Shifting our attention to the robustness and generalisability of learning from multimodal data, Chapter 5 explores domain generalisation. We propose Fish, a method that encourages models to leverage commonalities across multiple source domains for better generalisation to unseen target domains. By maximising the inter-domain gradient inner product, we achieve improved performance on a wide range of domain generalisation tasks.

Since the publication of this work, we have seen many useful applications of gradient alignment. Apart from concurrent work including Koyama and Yamaguchi (2021a) and Parascandolo et al. (2021) that also uses gradient alignment for domain generalisation, we have also seen its usage in federated learning (Dandi et al., 2022), semi-supervised learning (Jackson and Schulman, 2019) and zero-shot learning (Sariyildiz and Cinbis, 2019). Furthermore, examining gradient direction can also be used for dataset curation/condensation (Zhao et al., 2020; Killamsetty et al., 2021). Our approach Fish also directly inspired another method named Fishr (Rame et al., 2022) — instead of matching the domain-level gradients, Fishr matches the domain-level gradient variances and showed promising results in domain generalisation.

Masked image modelling Finally in Chapter 6, we examine SSL as a multimodal learning principle. To enhance the quality of representations learned from an image encoder, we propose a novel approach that involves masking out one augmented view using adversarially generated masks. This technique yields better quality representations by occluding complete semantic entities in the image. Similar to the framework used in our work, Yao et al. (2022) proposes MACRL which combines masked image modelling with contrastive learning. However, the more important contribution of this work is that it drew significant attention on the construction of mask for masked image modelling models. Inspired by our approach, Li et al. (2022b) proposes to use semantically guided mask by leveraging and refining the attention map of ViT (Li et al., 2022b). Prabhu et al. (2022) also uses attention-conditioned masking to perform visual domain adaptation tasks. Also similar to our work, Chen et al. (2023) finds that introducing object-aware priors when sampling masks can significantly improve performance.

In conclusion, this thesis has made significant contributions to the field of representation learning using multiple sources of data. Our proposed models and frameworks, such as MMVAE, contrastive learning for multimodal VAE, Fish, and ADIOS, have demonstrated improved performance, data efficiency, and generalisation capabilities. These findings contribute to the ongoing research efforts aimed at addressing the challenges of supervised learning and enhancing the robustness and accuracy of machine learning models in real-world applications.

While the main text of this thesis primarily focuses on multimodal learning itself, we also explored the generalisation of these methods to unseen distributions in deployment. To provide a comprehensive overview, we included this work in the appendix and directed interested readers to further explore this aspect.

7.2 Limitations

Robustness of Multimodal Learning This thesis discussed the robustness and generalisation performance in the context of discriminative models (Chapter 5, Chapter 6). Although both models showed improved performance on out-of-distribution generalisation by learning from multiple distributions, the robustness of learned representations was not discussed in the context of earlier chapters on generative models, i.e. multimodal VAEs. Given that past work has established clear theoretical connections between improved robustness and learning features that are stable across multiple distributions/environments (Arjovsky et al., 2019b; Sturma et al., 2023), it would be interesting to investigate how this manifests for multimodal deep generative models empirically.

Large scale experiments It is clear that Chapter 6 could have benefited from larger scale experiments. After all, one of the key benefits of SSL is enabling pretraining on a large amount of unlabelled data. To put this in more context, the largest dataset used in the experiments in Chapter 6 contains 100,000 images, and the most expressive model used, ResNet-50, contains 23 million parameters; in comparison, MAE (He et al., 2021) uses the ViT-Large architecture which contains

307 million parameters, and is pretrained on ImageNet-1k which contains more than 1 million images.

As a result our experiment results are not directly comparable to MAE’s — while this is indeed limited by the amount of computational resources available to the authors at the time, future work should look into extending the model to larger scale pretraining settings and compare the performance of this model to other state-of-the-art mask image modelling methods.

7.3 Future works

In addition to addressing the limitations described in the last section, there are a few other interesting directions that is worth exploring as extensions to this thesis.

Multimodal alignment of MIM and MLM In Chapter 6 we proposed ADIOS, which learns to mask out complete entities in images for mask image modelling (MIM). This is inspired by the fact that masks in mask language modelling (MLM) also tends to remove complete entities of the described scenes.

The question we could ask here is: can we enforce correspondence between the masked-out entity in language and vision, thereby learning better-aligned multimodal representations? One could consider learning an ADIOS-like model with the visual mask conditionally generated on the language mask, and supervise the learning of such correspondence through a cycle consistency objective.

Equivariance for SSL Traditional augmentation-based self-supervised learning (SSL) methods, e.g. SimCLR (Chen et al., 2020b), sets out to learn representations that are *invariant* to augmentations. Specifically, given a set of augmentations available T , methods like SimCLR augments the image into two different views by randomly sampling from T , and the objective is to minimise the distance between the representations of these two different views.

One of the main issues with augmentation-based SSL is that the model’s performance is highly sensitive to the choice of augmentations T , which can vary for different architectures and downstream tasks. In fact, the definition of “good augmentations”

has been a topic of discussion in many works (Tian et al., 2020a,c). For example, applying rotation can often harm the performance of the model as it alters the structural information of the data too drastically. Consequently, enforcing the representations of two rotated views to be the same can be too restrictive of a constraint.

To mitigate this issue, instead of aiming for invariance, we can consider making the model *equivariant* to image augmentations. This means learning a mapping function in the representation space that corresponds to the augmentations applied in the image space. By preserving the information about augmentations in the representations, we hope to make the model less sensitive to the choice of augmentations, allowing for a more diverse range of augmentations to be utilised. Additionally, utilising information about the specific augmentations applied to each image can serve as a useful signal for supervision and improve generalisation performance according to the InfoMax principle (Linsker, 1988).

While there are recent works that look into incorporating equivariance with SSL (Dangovski et al., 2022; Wang et al., 2021a), their experiments are conducted on a very limited set of augmentations, and typically enforces equivariance for one augmentation at each time. We hope to extend this to a more general set of augmentations, and potentially investigate the suitability of applying equivariance principle for “non-parametric augmentations” such as domain adaptation or learning from multimodal data.

Learning augmentations for SSL Another way to address the difficulty of selecting augmentations for SSL is to learn them jointly with the model. This could be done by adopting a similar training strategy to ADIOS, more specifically, learning a model that generates image augmentations adversarially to the image encoder. One could consider this as a more general form of masked image modelling, where we think of mask as a form of augmentation. The intuition of this draws parallels to Automatic Curriculum Learning (ACL) in Reinforcement Learning (RL), where the difficulty level of the environment that the agent is trained on is automatically

adjusted based on the agent performance itself. In RL, ACL is shown to have improved generalisation capability and is useful in the context of transfer learning and continual learning (Akkaya et al., 2019). By adopting similar principles in SSL, we are hoping to observe similar booster effect in the generalisation performance of the models.

Domain augmentations for generalisation As discussed in Chapter 5, many domain generalisation algorithms aim to learn only features in data that remains stable across multiple environments (i.e. invariant features), while discarding those that changes as the environment changes. To make this learning principle more explicit and interpretable, one could consider learning augmentations that makes the data domain agnostic. This can be done, again, in an adversarial manner, by utilising a domain classifier similar to Ganin et al. (2016b).

Bibliography

- S. M. Abd Elrahman and A. Abraham. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1(2013):332–340, 2013.
- S. Agarwal, G. Krueger, J. Clark, A. Radford, J. W. Kim, and M. Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021.
- I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.
- J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.
- A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.
- M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016.
- M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019a.
- M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019b.

S. Azizi, L. Culp, J. Freyberg, B. Mustafa, S. Baur, S. Kornblith, T. Chen, P. MacWilliams, S. S. Mahdavi, E. Wulczyn, et al. Robust and efficient medical imaging with self-supervision. *arXiv:2205.09723*, 2022.

A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. URL <https://ai.facebook.com/research/data2vec-a-general-framework-for-self-supervised-learning-in-speech-vision-and> Accessed: 2022-01-27.

Y. Balaji, S. Sankaranarayanan, and R. Chellappa. Metareg: towards domain generalization using meta-regularization. In *NIPS'18 Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, pages 1006–1016, 2018.

Y. Balaji, R. Chellappa, and S. Feizi. Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6500–6508, 2019.

H. Bao, L. Dong, and F. Wei. Beit: Bert pre-training of image transformers. *ArXiv preprint*, abs/2106.08254, 2021. URL <https://arxiv.org/abs/2106.08254>.

A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ArXiv preprint*, abs/2105.04906, 2021. URL <https://arxiv.org/abs/2105.04906>.

K. Barnard, P. Duygulu, D. Forsyth, N. d. Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of machine learning research*, 3(Feb): 1107–1135, 2003.

L. W. Barsalou. Grounded cognition. *Annual Review of Psychology*, 59:617–645, 2008.

- Z. Barzelay and Y. Y. Schechner. Harmony in motion. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- M. I. Bauer and P. N. Johnson-Laird. How diagrams can improve reasoning. *Psychological science*, 4(6):372–378, 1993.
- S. Beery, G. V. Horn, and P. Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 472–489, 2018.
- S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.
- Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- S. F. Bhat, I. Alhashim, and P. Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.
- D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003.
- S. L. Blodgett, L. Green, and B. T. O’Connor. Demographic dialectal variation in social media: A case study of african-american english. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, 2016.
- A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pages 19–26. Morgan Kaufmann, 2001.

- A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.
- P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016a.
- K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016b.
- W. Brendel and M. Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations (ICLR)*, 2019.
- J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah. Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4): 669–688, 1993.
- M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.
- Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2015.
- Y. Burda, R. B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

- M. C. Burkhart and K. Shan. Deep low-density separation for semi-supervised classification. In *International Conference on Computational Science*, pages 297–311, 2020.
- F. M. Carlucci, A. D’Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2229–2238, 2019.
- M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.
- M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html>.
- M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- M. Chatterjee and A. Cherian. Sound2sight: Generating visual dynamics from sound and context. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 701–719. Springer, 2020.
- H. Chen, W. Zhang, Y. Wang, and X. Yang. Improving masked autoencoders by learning where to mask. *arXiv preprint arXiv:2303.06583*, 2023.
- T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual*

- Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020a. URL <http://proceedings.mlr.press/v119/chen20j.html>.
- T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020b. URL <http://proceedings.mlr.press/v119/chen20j.html>.
- X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.
- X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.
- X. Chen, H. Fan, R. B. Girshick, and K. He. Improved baselines with momentum contrastive learning. *ArXiv preprint*, abs/2003.04297, 2020. URL <https://arxiv.org/abs/2003.04297>.
- X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.
- Z. Chen, Y. Luo, and N. Mesgarani. Deep attractor network for single-microphone speaker separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 246–250. IEEE, 2017.
- E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *The Journal of the acoustical society of America*, 25(5):975–979, 1953.
- C. Chu, A. Zhmoginov, and M. Sandler. Cyclegan, a master of steganography. *arXiv preprint arXiv:1712.02950*, 2017.
- J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Computer Vision—ACCV 2016 Workshops: ACCV 2016 International Workshops*,

- Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*, pages 251–263. Springer, 2017.
- C. Cremer, Q. Morris, and D. Duvenaud. Reinterpreting importance-weighted autoencoders. In *International Conference on Learning Representations (Workshop)*, 2017.
- C. Cremer, Q. Morris, and D. Duvenaud. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.
- E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 113–123. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00020. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.html.
- V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci. Solo-learn: A library of self-supervised methods for visual representation learning, 2021. URL <https://github.com/vturrisi/solo-learn>.
- V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 2022.
- Y. Dandi, L. Barba, and M. Jaggi. Implicit gradient alignment in distributed and federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6454–6462, 2022.
- R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljagic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022.

- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society series b-methodological*, 39(1):1–22, 1977.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- A. B. Dieng, D. Tran, R. Ranganath, J. W. Paisley, and D. M. Blei. Variational inference via χ upper bound minimization. In *31st Annual Conference on Neural Information Processing Systems, NIPS 2017*, pages 2732–2741, 2017.
- J. Dockès, G. Varoquaux, and J.-B. Poline. Preventing dataset shift from breaking machine-learning biomarkers. *GigaScience*, 2021.
- J. Dong, Z. Fang, A. Liu, G. Sun, and T. Liu. Confident anchor-induced multi-source free domain adaptation. *Advances in Neural Information Processing Systems*, 34: 2848–2860, 2021.
- A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.

- Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, volume 32, pages 6450–6461, 2019.
- M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL <https://openreview.net/forum?id=BkxfatVFWH>.
- M. Engelcke, O. P. Jones, and I. Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *ArXiv preprint*, abs/2104.09958, 2021. URL <https://arxiv.org/abs/2104.09958>.
- A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 3015–3024. PMLR, 2021. URL <http://proceedings.mlr.press/v139/ermolov21a.html>.
- J. E. Fan, D. Yamins, and N. B. Turk-Browne. Common object representations for visual recognition and production. *Cognitive Science*, 42:2670–2698, 2018.
- C. Fang, Y. Xu, and D. N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *2013 IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.
- C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML’17 Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 1126–1135, 2017.
- J. W. Fisher III, T. Darrell, W. Freeman, and P. Viola. Learning joint statistical models for audio-visual fusion and segregation. *Advances in neural information processing systems*, 13, 2000.

- K. Frans. Variational autoencoders explained, 2016. URL <http://kvfrans.com/variational-autoencoders-explained/>.
- C. Gan, D. Huang, P. Chen, J. B. Tenenbaum, and A. Torralba. Foley music: Learning to generate music from videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 758–775. Springer, 2020.
- Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1180–1189, 2015.
- Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016a.
- Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016b.
- Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016c.
- W. V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. V. Gool. Scan: Learning to classify images without labels. In *ECCV (10)*, pages 268–285, 2020.
- R. Gao and K. Grauman. 2.5 d visual sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2019.
- R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2018.

- R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.
- Z. Ghahramani and M. Jordan. Factorial hidden markov models. *Advances in Neural Information Processing Systems*, 8, 1995.
- S. Ghalebikesabi, R. Cornish, C. Holmes, and L. J. Kelly. Deep generative missingness pattern-set mixture models. In *AISTATS*, pages 3727–3735, 2021.
- M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2551–2559, 2015.
- S. S. Ghosal, Y. Ming, and Y. Li. Are vision transformers robust to spurious correlations? *arXiv:2203.09125*, 2022.
- B. Glocker, R. Robinson, D. C. Castro, Q. Dou, and E. Konukoglu. Machine learning with multi-site imaging data: an empirical study on the impact of scanner effects. *arXiv:1910.04597*, 2019.
- Y. Gong, H. Hajimirsadeghi, J. He, T. Durand, and G. Mori. Variational selective autoencoder: Learning from partially-observed heterogeneous data. In *AISTATS*, pages 2377–2385, 2021.
- K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2424–2433. PMLR, 2019. URL <http://proceedings.mlr.press/v97/greff19a.html>.

- A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19:513–520, 2006.
- A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. *Covariate shift and local learning by distribution matching*, pages 131–160. MIT Press, Cambridge, MA, USA, 2009a.
- A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4): 5, 2009b.
- J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - A new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html>.
- G. Grimmett, G. R. Grimmett, D. Stirzaker, et al. *Probability and random processes*. Oxford university press, 2001.
- I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.
- I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. *arXiv:2007.01434*, 2020.
- W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

- R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.
- S. Harary, E. Schwartz, A. Arbelle, P. Staar, S. Abu-Hussein, E. Amrani, R. Herzig, A. Alfassy, R. Giryes, H. Kuehne, et al. Unsupervised domain generalization by learning a bridge across domains. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- T. B. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938, 2018.
- R. Hataya, J. Zdenek, K. Yoshizoe, and H. Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *16th European Conference on Computer Vision, ECCV 2020*, pages 1–16, 2020.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020a. doi: 10.1109/CVPR42600.2020.00975. URL <https://doi.org/10.1109/CVPR42600.2020.00975>.
- K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020b. doi: 10.1109/CVPR42600.2020.00975. URL <https://doi.org/10.1109/CVPR42600.2020.00975>.
- K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *ArXiv*, abs/2111.06377, 2021.

- O. J. Hénaff. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4182–4192. PMLR, 2020. URL <http://proceedings.mlr.press/v119/henaff20a.html>.
- J. Henrich, S. J. Heine, and A. Norenzayan. Most people are not weird. *Nature*, 2010.
- K. Hermann and A. Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Conference on Neural Information Processing Systems*, 2020.
- J. Hershey and J. Movellan. Audio vision: Using audio-visual synchrony to locate sounds. *Advances in neural information processing systems*, 12, 1999.
- J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 31–35. IEEE, 2016.
- I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002a.
- G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002b.
- R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations*,

- ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=Bklr3j0cKX>.
- M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- G. V. Horn, O. M. Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8769–8778. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00914. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Van_Horn_The_INaturalist_Species_CVPR_2018_paper.html.
- C.-C. Hsu, C.-W. Lin, W.-T. Su, and G. Cheung. Sigan: Siamese generative adversarial network for identity-preserving face hallucination. *IEEE Transactions on Image Processing*, 28(12):6225–6236, 2019.
- W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers. In *International Conference on Machine Learning*, pages 2029–2037, 2018a.
- W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers. In *International Conference on Machine Learning*, 2018b.
- W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.
- G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

- Z. Huang, H. Wang, E. P. Xing, and D. Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pages 124–140, 2020.
- A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.
- A. Hyvarinen and H. Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR, 2017.
- M. Ilse, J. Tomczak, and P. Forré. Selecting data augmentation for simulating interventions. In *ICML 2021: 38th International Conference on Machine Learning*, pages 4555–4562, 2021.
- O. Ivanov, M. Figurnov, and D. P. Vetrov. Variational autoencoder with arbitrary conditioning. In *International Conference on Learning Representations*, pages 1–25, 2019.
- J. Jackson and J. Schulman. Semi-supervised learning by label gradient alignment. *arXiv preprint arXiv:1902.02336*, 2019.
- D. Jeong, S. Doh, and T. Kwon. Träumerai: Dreaming music with stylegan. *arXiv preprint arXiv:2102.04680*, 2(4):10, 2021.
- C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021a.
- C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021b.

- T. Joachims. Transductive inference for text classification using support vector machines. In *ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, 1999.
- J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.215. URL <https://doi.org/10.1109/CVPR.2017.215>.
- T. Joy, S. M. Schmon, P. H. Torr, N. Siddharth, and T. Rainforth. Capturing label characteristics in vaes. *arXiv preprint arXiv:2006.10102*, 2020.
- N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 212–217. Association for Computational Linguistics, 2014.
- D. Kalimeris, G. Kaplun, P. Nakkiran, B. Edelman, T. Yang, B. Barak, and H. Zhang. Sgd on neural networks learns functions of increasing complexity. *Conference on Neural Information Processing Systems*, 2019.
- B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.
- K. Killamsetty, S. Durga, G. Ramakrishnan, A. De, and R. Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021.
- B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim. Learning not to learn: Training deep neural networks with biased data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

- D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2014.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014a.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014b.
- P. Kirichenko, P. Izmailov, and A. G. Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv:2204.02937*, 2022.
- A. S. Koepke, O. Wiles, Y. Moses, and A. Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1838–1842. IEEE, 2020.
- P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.
- P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. Haque, S. M. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, 2021.
- M. Koyama and S. Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. In *arXiv e-prints*, 2021a.
- M. Koyama and S. Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. In *arXiv e-prints*, 2021b.

- M. Koyama, K. Minami, T. Miyato, and Y. Gal. Contrastive representation learning with trainable augmentation channel. *ArXiv preprint*, abs/2111.07679, 2021. URL <https://arxiv.org/abs/2111.07679>.
- P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 109–117, 2011. URL <https://proceedings.neurips.cc/paper/2011/hash/beda24c1e1b46055dff2c39c98fd6fc1-Abstract.html>.
- A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 2019.
- T. A. Le, M. Igl, T. Rainforth, T. Jin, and F. Wood. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.
- R. Le Bras, S. Swayamdipta, C. Bhagavatula, R. Zellers, M. Peters, A. Sabharwal, and Y. Choi. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, 2020.
- G. Le Moing, J. Ponce, and C. Schmid. Ccvs: context-aware controllable video synthesis. *Advances in Neural Information Processing Systems*, 34:14042–14055, 2021.
- C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017.

- T.-W. Lee. Independent component analysis. In *Independent component analysis*, pages 27–66. Springer, 1998.
- C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.
- D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551, 2017.
- D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.
- D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. Hospedales. Episodic training for domain generalization. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1446–1455, 2019.
- F. Li, H. Zhang, S. Liu, L. Zhang, L. M. Ni, H.-Y. Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022a.
- G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*, 2022b.
- H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018b.
- J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

- S. Li, C. Liu, Q. Lin, B. Xie, Z. Ding, G. Huang, and J. Tang. Domain conditioned adaptation network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11386–11393, 2020.
- Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 647–663, 2018.
- R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.
- Z. C. Lipton. The mythos of model interpretability. *ACM Queue*, 61(10):36–43, 2018.
- H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv:2110.05025*, 2021.
- M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017.
- M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. Few-shot unsupervised image-to-image translation, 2019.
- Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022a.
- Z. Liu, H. Mao, W. Chao-Yuan, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *arXiv preprint arXiv: 2201.03545*, 2022b.
- Z. Liu, Y. Xu, Y. Xu, Q. Qian, H. Li, R. Jin, X. Ji, and A. B. Chan. An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation. *arXiv:2205.12753*, 2022c.

- M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 97–105, 2015.
- M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016a.
- M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016b.
- D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 6467–6476, 2017.
- X. Luo, L. Wei, L. Wen, J. Yang, L. Xie, Z. Xu, and Q. Tian. Rectifying the shortcut learning of background for few-shot learning. *Conference on Neural Information Processing Systems*, 2021.
- C. Ma, S. Tschitschek, K. Palla, J. M. Hernández-Lobato, S. Nowozin, and C. Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243, 2018.
- E. Marcheret, G. Potamianos, J. Vopicka, and V. Goel. Detecting audio-visual synchrony using deep neural networks. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- D. Massiceti, P. K. Dokania, N. Siddharth, and P. H. Torr. Visual dialogue without vision or dialogue. In *NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018a.
- D. Massiceti, N. Siddharth, P. K. Dokania, and P. H. Torr. FlipDial: a generative model for two-way visual dialogue. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

- E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In K. Chaudhuri and R. Salakhutdinov, editors, *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 4402–4412, Long Beach, California, USA, June 2019a. PMLR.
- E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019b.
- P.-A. Mattei and J. Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, pages 4413–4423, 2019.
- L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>, 2017.
- D. F. McAllister, R. D. Rodman, D. L. Bitzer, and A. S. Freeman. Lip synchronization of speech. In *Audio-Visual Speech Processing: Computational & Cognitive Science Approaches*, 1997.
- A. K. Menon, A. S. Rawat, and S. Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2020.
- D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in Neural Information Processing Systems 9*, pages 571–577, 1996.
- K. Minoura, K. Abe, H. Nam, H. Nishikawa, and T. Shimamura. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. *Cell reports methods*, 1(5):100071, 2021.
- A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q.

- Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2265–2273, 2013. URL <https://proceedings.neurips.cc/paper/2013/hash/db2b4182156b2f1f817860ac9f409ad7-Abstract.html>.
- S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte carlo gradient estimation in machine learning. *arXiv preprint arXiv:1906.10652*, 2019.
- S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. Unified deep supervised domain adaptation and generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5716–5726, 2017.
- T. Mukherjee, M. Yamada, and T. M. Hospedales. Deep matching autoencoders. *CoRR*, abs/1711.06047, 2017.
- A. Nazábal, P. M. Olmos, Z. Ghahramani, and I. Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.
- J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 689–696, 2011.
- V.-A. Nguyen, T. Nguyen, T. Le, Q. H. Tran, and D. Phung. Stem: An approach to multi-source domain adaptation with guarantees. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9352–9363, 2021.
- A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv: Learning*, 2018.
- K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134, 2000.

- M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.
- A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.
- G. Pandey and A. Dukkipati. Variational methods for conditional multimodal deep learning. In *International Joint Conference on Neural Networks*, pages 308–315. IEEE, 2017.
- G. Parascandolo, A. Neitz, A. Orvieto, L. Gresele, and B. Schölkopf. Learning explanations that are hard to vary. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.
- G. Y. Park and S. W. Lee. Information-theoretic regularization for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9214–9223, 2021.
- D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.278. URL <https://doi.org/10.1109/CVPR.2016.278>.
- Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019.
- N.-Q. Pham, G. Kruszewski, and G. Boleda. Convolutional neural network language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1153–1162, 2016.

- V. Prabhu, S. Yenamandra, A. Singh, and J. Hoffman. Adapting self-supervised vision transformers by probing attention-conditioned masking consistency. *arXiv preprint arXiv:2206.08222*, 2022.
- Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 2352–2360. Curran Associates, Inc., 2016.
- J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
- R. Q. Quiroga, A. Kraskov, C. Koch, and I. Fried. Explicit encoding of multimodal percepts by single neurons in the human brain. *Current Biology*, 19(15):1308–1313, 2009.
- A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- T. Rainforth, A. R. Kosiorek, T. A. Le, C. J. Maddison, M. Igl, F. Wood, and Y. W. Teh. Tighter variational bounds are not necessarily better. *International Conference on Machine Learning (ICML)*, 2018.
- A. Rame, C. Dancette, and M. Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.
- A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine*

- Learning Research*, pages 8821–8831. PMLR, 2021. URL <http://proceedings.mlr.press/v139/ramesh21a.html>.
- S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.
- S. E. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. *CoRR*, abs/1605.05395, 2016.
- M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343, 2018.
- C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.
- J. Robinson, L. Sun, K. Yu, K. Batmanghelich, S. Jegelka, and S. Sra. Can contrastive learning avoid shortcut solutions? *Conference on Neural Information Processing Systems*, 2021.
- G. Roeder, Y. Wu, and D. K. Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6925–6934. Curran Associates, Inc., 2017.
- M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. A causal perspective on domain adaptation. *stat*, 1050:19, 2015.
- O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- S. Roweis. One microphone source separation. *Advances in neural information processing systems*, 13, 2000.

- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.
- S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.
- S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.
- V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- M. B. Sariyildiz and R. G. Cinbis. Gradient matching generative networks for zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2168–2178, 2019.
- S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han. Learning to optimize domain specific normalization for domain generalization. In *European Conference on Computer Vision*, pages 68–83, 2019.
- H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. In *Conference on Neural Information Processing Systems*, 2020.
- J. Shen, X. Zhen, M. Worring, and L. Shao. Variational multi-task learning with gumbel-softmax priors. *Advances in Neural Information Processing Systems*, 34: 21031–21042, 2021.
- Y. Shi. How i learned to stop worrying and write elbo (and its gradients) in a billion ways, 2020. URL <https://yugeten.github.io/posts/2020/06/elbo/>.

- Y. Shi. An incomplete and slightly outdated literature review on augmentation based self-supervise learning, 2021. URL <https://yugeten.github.io/posts/2021/12/ss1/>.
- Y. Shi, S. Narayanaswamy, B. Paige, and P. H. S. Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15692–15703, 2019a. URL <https://proceedings.neurips.cc/paper/2019/hash/0ae775a8cb3b499ad1fca944e6f5c836-Abstract.html>.
- Y. Shi, B. Paige, P. Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems*, 32, 2019b.
- Y. Shi, N. Siddharth, B. Paige, and P. H. Torr. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *arXiv*, (NeurIPS), 2019c. ISSN 23318422. URL <https://arxiv.org/pdf/1911.03393.pdf>.
- Y. Shi, B. Paige, P. H. Torr, and N. Siddharth. Relating by contrasting: A data-efficient framework for multimodal generative models. *arXiv preprint arXiv:2007.01179*, 2020.
- Y. Shi, J. Seely, P. H. S. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.
- Y. Shi, I. Daunhawer, J. E. Vogt, P. H. Torr, and A. Sanyal. How robust are pre-trained models to distribution shift? *arXiv preprint arXiv:2206.08871*, 2022a.
- Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022b.

- Y. Shi, N. Siddharth, P. Torr, and A. R. Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, pages 20026–20040. PMLR, 2022c.
- C. Silberer and M. Lapata. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 721–732, 2014.
- J. M. Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8(5):371–391, Sep 1994.
- K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.
- J. Son Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6447–6456, 2017.
- H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4004–4012. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.434. URL <https://doi.org/10.1109/CVPR.2016.434>.
- B. E. Stein, T. R. Stanford, and B. A. Rowland. The neural basis of multisensory integration in the midbrain: its organization and maturation. *Hearing research*, 258(1-2):4–15, 2009.
- P. Stock and M. Cisse. Convnets and imagenet beyond accuracy: Explanations, bias detection, adversarial examples and model criticism. *arXiv preprint arXiv:1711.11443*, 2017.
- N. Sturma, C. Squires, M. Drton, and C. Uhler. Unpaired multi-domain causal representation learning. *arXiv preprint arXiv:2302.00993*, 2023.

- B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016a.
- B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016b.
- T. M. Sutter, I. Daunhawer, and J. E. Vogt. Multimodal generative learning utilizing jensen-shannon divergence. In *Workshop on Visually Grounded Interaction and Language at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, volume 33, pages 6100–6110, 2020.
- T. M. Sutter, I. Daunhawer, and J. E. Vogt. Generalized multimodal elbo. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.
- M. Suzuki, K. Nakayama, and Y. Matsuo. Joint multimodal learning with deep generative models. In *International Conference on Learning Representations Workshop*, 2017.
- C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations*, 2017.
- A. Tamkin, M. Wu, and N. D. Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=enovQWLsfyL>.
- R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

- A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *ICLR (Workshop)*, 2017.
- R. Tatman. Gender and dialect bias in youtube’s automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59, 2017.
- D. Teney, E. Abbasnejad, S. Lucey, and A. van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16761–16772, 2022.
- S. Thrun and L. Pratt, editors. *Learning to Learn*. Kluwer Academic Publishers, USA, 1998. ISBN 0792380479.
- Y. Tian and J. Engel. Latent translation: Crossing modalities by bridging generative models. *arXiv preprint arXiv:1902.08261*, 2019.
- Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020a.
- Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020b.
- Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020c.
- A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011a.
- A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR ’11*,

- page 1521–1528, USA, 2011b. IEEE Computer Society. ISBN 9781457703942. doi: 10.1109/CVPR.2011.5995347. URL <https://doi.org/10.1109/CVPR.2011.5995347>.
- H. Touvron, M. Cord, A. El-Nouby, P. Bojanowski, A. Joulin, G. Synnaeve, and H. Jégou. Augmenting convolutional networks with attention-based aggregation. *ArXiv preprint*, abs/2112.13692, 2021. URL <https://arxiv.org/abs/2112.13692>.
- Y. H. Tsai, P. P. Liang, A. A. Bagherzade, L.-P. Morency, and R. Salakhutdinov. Learning factorized multimodal representations. In *International Conference on Learning Representations*, 2019.
- G. Tucker, D. Lawson, S. Gu, and C. J. Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. In *International Conference on Learning Representations*, 2019.
- E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.
- A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.
- R. Vedantam, I. Fischer, J. Huang, and K. Murphy. Generative models of visually grounded imagination. In *International Conference on Learning Representations*, 2018.

- H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5385–5394, 2017.
- T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.
- R. Volpi and V. Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7980–7989, 2019.
- C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- H. Wang, S. Ge, Z. C. Lipton, and E. P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, pages 10506–10518, 2019.
- J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.
- T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR, 2020. URL <http://proceedings.mlr.press/v119/wang20k.html>.
- W. Wang, X. Yan, H. Lee, and K. Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016.

- X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.
- Y. Wang, Z. Geng, F. Jiang, C. Li, Y. Wang, J. Yang, and Z. Lin. Residual relaxation for multi-view representation learning. *Advances in Neural Information Processing Systems*, 34:12104–12115, 2021a.
- Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021b.
- K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 1473–1480, 2005. URL <https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html>.
- P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010a.
- P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010b.
- F. Wu and X. Zhuang. Unsupervised domain adaptation with variational approximation for cardiac segmentation. *IEEE Transactions on Medical Imaging*, 40(12):3555–3567, 2021.
- M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2018a.

- M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning. *Adv. Neural Inf. Process. Syst.*, 2018-Decem(Nips):5575–5585, 2018b. ISSN 10495258. URL <https://arxiv.org/pdf/1802.05335.pdf>.
- K. Y. Xiao, L. Engstrom, A. Ilyas, and A. Madry. Noise or signal: The role of image backgrounds in object recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=gl3D-xY7wLq>.
- H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.
- S. Yan, H. Song, N. Li, L. Zou, and L. Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.
- W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021.
- X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.
- Y. Yao, N. Desai, and M. Palaniswami. Masked contrastive representation learning. *arXiv preprint arXiv:2211.06012*, 2022.
- F. Ye and A. G. Bors. Lifelong mixture of variational autoencoders. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- G. Yildirim, N. Jetchev, and U. Bergmann. Unlabeled disentangling of gans with guided siamese networks. 2018.
- I. Yildirim. From perception to conception: Learning multisensory representations. 2014.
- I. Yildirim. From perception to conception: learning multisensory representations. 2014.

- D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245. IEEE, 2017.
- J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.
- L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.
- J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 2021. URL <http://proceedings.mlr.press/v139/zbontar21a.html>.
- X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.
- C. Zhang, Z. Yang, X. He, and L. Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–1, 2020.
- X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu. Towards unsupervised domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4910–4920, 2022.
- B. Zhao, K. R. Mopuri, and H. Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

- S. Zhao, J. Song, and S. Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.
- D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 321–328. MIT Press, 2003. URL <https://proceedings.neurips.cc/paper/2003/hash/87682805257e619d49b8e0dfdc14affa-Abstract.html>.
- J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *ArXiv preprint*, abs/2111.07832, 2021. URL <https://arxiv.org/abs/2111.07832>.
- K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578, 2020.
- Q. Zhou, K.-Y. Zhang, T. Yao, R. Yi, S. Ding, and L. Ma. Adaptive mixture of experts learning for generalizable face anti-spoofing. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6009–6018, 2022.
- Z.-H. Zhou and M. Li. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.
- Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439, 2010.
- J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017a.

- J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017b.
- X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In T. Fawcett and N. Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 912–919. AAAI Press, 2003. URL <http://www.aaai.org/Library/ICML/2003/icml03-118.php>.

Appendices



How Robust is Unsupervised
Representation Learning to Distribution
Shift?

Abstract

The robustness of machine learning algorithms to distributions shift is primarily discussed in the context of supervised learning (SL). As such, there is a lack of insight on the robustness of the representations learned from unsupervised methods, such as SSL and Autoencoder (AE), to distribution shift. We posit that the *input-driven* objectives of unsupervised algorithms lead to representations that are more robust to distribution shift than the *target-driven* objective of SL. We verify this by extensively evaluating the performance of SSL and AE on both synthetic and realistic distribution shift datasets. Following observations that the linear layer used for classification itself can be susceptible to spurious correlations, we evaluate the representations using a linear head trained on a small amount of out-of-distribution (OOD) data, to isolate the robustness of the learned representations from that of the linear head. We also develop “controllable” versions of existing realistic domain generalisation datasets with adjustable degrees of distribution shifts. This allows us to study the robustness of different learning algorithms under versatile yet realistic distribution shift conditions. Our experiments show that representations learned from unsupervised learning algorithms generalise better than SL under a wide variety of extreme as well as realistic distribution shifts.

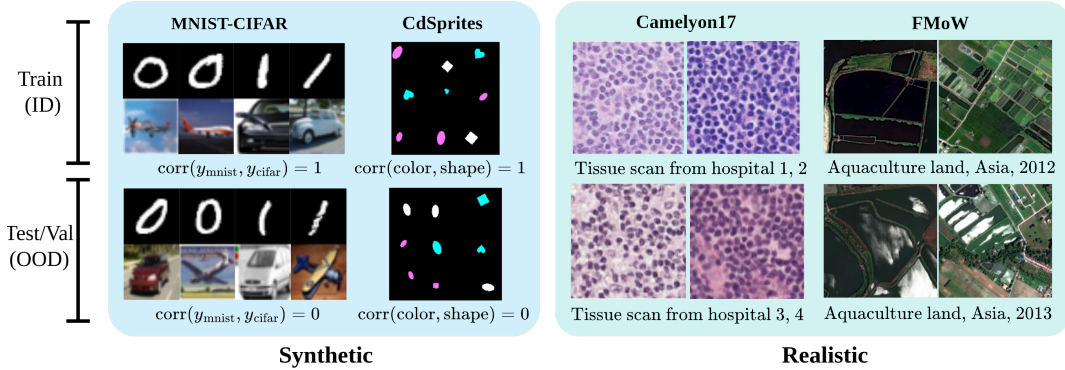


Figure A.1: Synthetic vs. realistic distribution shift: The distribution shift in synthetic datasets (*left*, MNIST-CIFAR and CdSprites) are usually extreme and controllable (adjusted via changing the correlation); for realistic datasets (*right*, WILDS-Camelyon17 and FMoW) distribution shift can be subtle, hard to identify and impossible to control.

A.1 Introduction

Machine Learning (ML) algorithms are classically designed under the statistical assumption that the training and test data are drawn from the same distribution. However, this assumption does not hold in most cases of real world deployment of ML systems. For example, medical researchers might obtain their training data from hospitals in Europe, but deploy their trained models in Asia; the changes in conditions such as imaging equipment and demography result in a shift in the data distribution between train and test set (Dockès et al., 2021; Glocker et al., 2019; Henrich et al., 2010). To perform well on such tasks requires the models to generalise to unseen distributions — an important property that is not evaluated on standard machine learning datasets like ImageNet, where the train and test set are sampled i.i.d. from the same distribution.

With increasing attention on this issue, researchers have been probing the generalisation performance of ML models by creating datasets that feature distribution shift tasks (Koh et al., 2021; Gulrajani and Lopez-Paz, 2020; Shah et al., 2020) and proposing algorithms that aim to improve generalisation performance under distribution shift (Ganin et al., 2016b; Arjovsky et al., 2019a; Sun and Saenko, 2016a; Sagawa et al., 2020; Shi et al., 2022b). In this work, we identify three specific problems with current approaches in distribution shift problems, in computer vision, and develop a suite of experiments to address them.

A.1.1 Existing problems and Contributions

Problem 1: The outdated focus on supervised regime for distribution shift

In ML research, distribution shift has been studied in various contexts under different terminologies such as simplicity bias (Shah et al., 2020), dataset bias (Torralba and Efros, 2011a), shortcut learning (Geirhos et al., 2020), and domain adaptation and generalisation (Koh et al., 2021; Gulrajani and Lopez-Paz, 2020). Most of these work are carried out under the scope of supervised learning (SL), including various works that either investigate spurious correlations (Shah et al., 2020; Hermann and Lampinen, 2020; Kalimeris et al., 2019) or those that propose specialised methods to improve generalisation and/or avoid shortcut solutions (Arjovsky et al., 2019a; Ganin et al., 2016b; Sagawa et al., 2020; Teney et al., 2022). However, recent research (Shah et al., 2020; Geirhos et al., 2020) highlighted the extreme vulnerability of SL methods to spurious correlations: they are susceptible to learning only features that are irrelevant to the true labelling functions yet highly predictive of the labels. This behaviour is not surprising given SL’s *target-driven* objective: when presented with two features that are equally predictive of the target label, SL models have no incentive to learn both as learning only one of them suffices to predict the target label. This leads to poor generalisation when the learned feature is missing in the OOD test set.

On the other hand, in recent times, research in computer vision has seen a surge of unsupervised representation learning algorithms. These include self-supervised learning (SSL) algorithms (e.g., Chen et al. (2020a); Grill et al. (2020); Chen and He (2021)), which learn representations by enforcing invariance between the representations of two distinctly augmented views of the same image, and Autoencoder (AE) (Rumelhart et al., 1985; Kingma and Welling, 2014a; Higgins et al., 2017; Burda et al., 2016), which learn representations by reconstructing the input image. The immense popularity of these methods are mostly owed to their impressive performance on balanced in-distribution (ID) test datasets — how they perform on distribution shift tasks remains largely unknown. However,

in distribution shift tasks, it is particularly meaningful to study unsupervised algorithms. This is because, in comparison to SL, their learning objectives are more *input-driven* i.e. they are incentivised to learn representations that most accurately represent the input data (Chen et al., 2020a; Alemi et al., 2017). When presented with two features equally predictive of the labels, unsupervised learning algorithms encourage the model to go beyond learning what’s enough to predict the label, and instead focus on maximising the mutual information between the learned representations and the input. We hypothesise that this property of unsupervised representation learning algorithms helps them avoid the exploitation of spurious correlations, and thus fare better under distribution shift, compared to SL.

Contribution: *Systematically evaluate SSL and AE on distribution shift tasks.* We evaluate and compare the generalisation performance of unsupervised representation learning algorithms, including SSL and AE, with standard supervised learning. See appendix A.2 for more details on our experiments.

Problem 2: Disconnect between synthetic and realistic datasets Broadly speaking, there exists two types of datasets for studying distribution shift: synthetic datasets where the shift between train/test distribution is explicit and controlled (e.g. MNIST-CIFAR (Shah et al., 2020), CdSprites (Shi et al., 2022b)) and realistic datasets featuring implicit distribution shift in the real world (e.g. WILDS (Koh et al., 2021)). We provide visual examples in fig. A.1.

Synthetic datasets allow for explicit control of the distribution shift and are, thus, an effective diagnostic tool for generalisation performance. However, the simplistic nature of these datasets poses concerns about the generality of the findings drawn from these experiments; a model’s robustness to spurious correlation on certain toy datasets is not very useful if it fails when tested on similar real-world problems. On the other hand, realistic datasets often feature distribution shifts that are subtle and hard to define (see fig. A.1, right). As a result, generalisation performances of different algorithms tend to fluctuate across datasets (Koh et al., 2021; Gulrajani and Lopez-Paz, 2020) with the cause of said fluctuations remaining unknown.

A. How Robust is Unsupervised Representation Learning to Distribution Shift?161

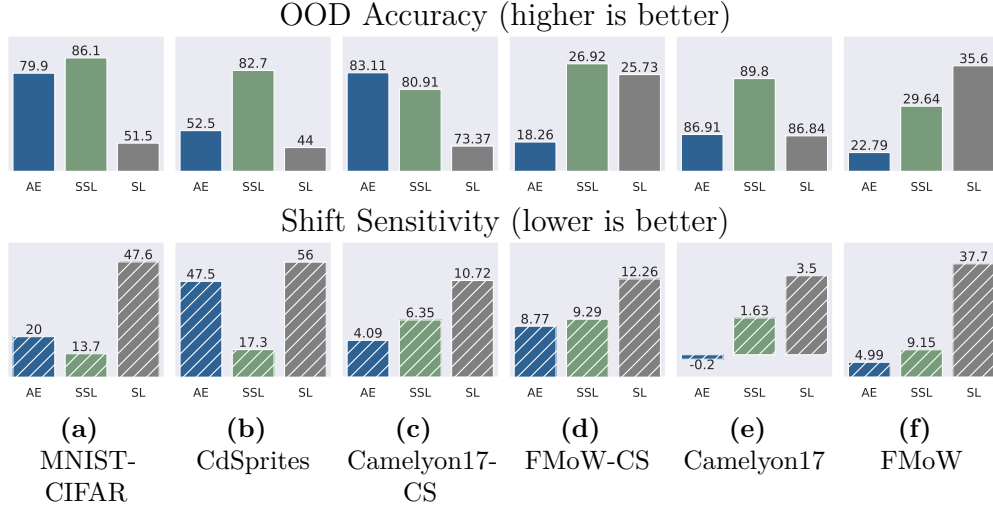


Figure A.2: Performance of auto-encoder (AE), self-supervised learning (SSL), supervised learning (SL) models. **Top row:** the OOD test set accuracy (%) using linear heads trained on OOD data; **Bottom row:** Shift Sensitivity (see appendix A.2 for definition), measures models’ sensitivity to distribution shift. Note here $r_{id} = 1$ for Camelyon17-CS, FMoW-CS, and CdSprites (see appendices A.3.1 and A.3.2).

Contribution: *Controllable but realistic distribution shift tasks.* In addition to evaluating models on both synthetic and realistic datasets, we subsample realistic domain generalisation datasets in WILDS to artificially inject explicit spurious correlations between domains and labels. This allows us to directly control the level of shift in these realistic distribution shift datasets. We refer to them as *Controllable-Shift (CS) datasets*.¹

Problem 3: The linear classifier head strongly biases the evaluation protocol The most popular evaluation protocol for representation learning algorithms (both supervised and unsupervised) is *linear probing*. This involves freezing the weights of the representation learning model and training a linear classifier head on top of that to predict the labels. For distribution shift problems, this linear classifier is typically trained on the same training set as the representation learning backbone. Kang et al. (2020); Menon et al. (2020) observed the interesting phenomenon that this final layer linear classifier can be extremely susceptible to spurious correlations, causing poor OOD performance. Under simple, synthetic set up, they showed

¹While datasets like ImageNet-9 and SVSF also provides distribution shift controls for real images, our sub-sampling of realistic domain generalisation datasets allows us to gain such control *cheaply* on a *large* set of datasets, and in addition provide further analysis on models’ performance on *existing benchmarks* (WILDS).

that SL models’ performance on OOD test set can be dramatically improved by simply retraining the linear classifier on data where the spurious correlation is absent. This indicates that the linear classifier can be a strong source of model bias in distribution shift tasks, and to disentangle the linear classifier bias from the generalisation performance of the learned representations, it is advisable to re-train the linear head on OOD data during evaluation. Note that although retraining linear classifier head is already standard practice in transfer learning, its application is necessary as the pre-training task and target task are typically different; on the other hand, retraining linear head is neither necessary nor standard practice in distribution shift problems, despite the recognition of linear head bias in recent work (Kang et al., 2020; Menon et al., 2020).

Contribution: *OOD linear head.* When reporting OOD accuracy, we use a linear head trained on small amount of left-out OOD data as opposed to ID data, as is standard practice. This allows us to isolate the bias of the linear head from the generalisability of learned representations. We also quantify the linear head bias to highlight the importance of this treatment. With these results, we wish to establish OOD linear head evaluation as a standard protocol for evaluating robustness of representation learning algorithms to distribution shift.

In summary, we develop a suite of experiments and datasets to evaluate the performance of various representation learning paradigms under distribution shift. Figure A.2 provides a summary of our results, comparing a range of methods from the following classes of algorithms: I) SSL, II) AE and III) SL. Note that though the intuition that unsupervised objectives should be better at distribution shift tasks is well-established in theory (Chen et al., 2020a; Alemi et al., 2017), state-of-the-art methods are predominantly developed under SL. To the best of our knowledge, we are the first to systematically evaluate and compare unsupervised representation learning methods to SL under distribution shift. The models are evaluated on both synthetic and realistic distribution shift datasets. Further, the models are also

evaluated on CS datasets that contains controllable, explicit spurious correlations in realistic datasets. The main takeaways from this paper are:

- **SSL and AE are more robust than SL to extreme distribution shift:** Figures A.2a to A.2d shows results on distribution shift scenarios where the training set encodes extreme spurious correlations. In this setting, for both synthetic (figs. A.2a and A.2b) and real world (figs. A.2c and A.2d) datasets, SSL and AE consistently outperforms SL in terms of OOD accuracy (top row);
- **Compared to SL, SSL and AE’s performance drop less under distribution shift:** The bottom row of fig. A.2 compares the shift sensitivity (s) of different models (see appendix A.2 for definition). Smaller s is desirable as it indicates lower sensitivity to distribution shift. Results show that SSL and AE algorithms are significantly more stable under distribution shift than SL;
- **Generalisation performance on distribution shift tasks can be significantly improved by retraining the linear head:** We show a large performance boost for all models, when evaluated using linear head trained on a small amount of OOD data, in contrast to the baseline linear head trained on ID data. The surprising gain of this cheap procedure, even on realistic problems, highlights the importance of isolating the linear head bias when evaluating generalisation performance.

A.2 Setting Up

In appendix A.1 we identified three problems in the existing literature that we wish to address in this work. In this section, we will introduce the necessary experimental set-up in further details. In brief, we compare eight ML algorithms on six datasets using three relevant metrics.

Algorithms: $\times 3$ SSL, $\times 4$ AE, $\times 1$ SL. We compare seven unsupervised representation learning algorithms against SL, including three SSL algorithms 1) SimCLR

(Chen et al., 2020a), 2) SimSiam (Chen and He, 2021), and 3) BYOL (Grill et al., 2020); and four AE algorithms 1) Autoencoder (Rumelhart et al., 1985), 2) VAE (Kingma and Welling, 2014a), 3) β -VAE (Higgins et al., 2017) and 4) IWAE (Burda et al., 2016). These popular methods in SSL and latent generative models have not yet been systematically evaluated under distribution shift tasks prior to our work. We compare the performance of these models against a standard supervised learning (SL) algorithm used as a representation learning model.

Datasets: $\times 2$ *synthetic*, $\times 2$ *realistic*, $\times 2$ *controllable shift*. We evaluate our models on two synthetic datasets, namely MNIST-CIFAR (Shah et al. (2020); see appendix A.3.1) and CdSprites (Shi et al. (2022b); see appendix A.3.1), as well as two realistic datasets from WILDS (Koh et al., 2021): Camelyon17 and FMoW (see appendix A.3.2). However, as mentioned in appendix A.1.1, both the synthetic and the realistic datasets have their own drawbacks. To further understand the models’ performance and draw conclusions that are generalisable, we also provide a framework for creating *controllable shift* datasets from realistic datasets like those in WILDS, by subsampling the data to introduce spurious correlations between the domain and label information in the training set. Changing this correlation varies the degree of distribution shift between the (ID) train and (OOD) test split, which allows us to analyse the models’ performance more effectively under realistic, yet controllable, distribution shift. We refer to this controllable shift versions of the two datasets Camelyon17-CS and FMoW-CS, and provide further details on the datasets in appendix A.3.2.

Evaluation: 3 *metrics*. Before discussing our proposed metrics, we first define some necessary notations. We separate a model trained to perform a classification task into two parts, namely, 1) backbone f , denoting the part of the model that generates representations from the data, and 2) final linear head c , which takes the representations from f and outputs the final prediction. Further, we refer to the final linear head trained on representations from the ID train set as c_i , and that trained on representations from the OOD test set as c_o . Since the backbone f is

always trained on the ID train set, we do not make any notation distinction on its training distribution. We also denote the accuracy of f and c on the ID test data as $\text{acc}_i(f, c)$, and on the OOD test data as $\text{acc}_o(f, c)$.

As noted in appendix A.1.1, we report the OOD accuracy of the algorithms using linear heads trained on OOD data (instead of those trained on ID data as per standard practice), i.e. $\text{acc}_o(f, c_o)$. This is necessary to disentangle the bias of the linear head from that of the representations. To highlight the importance of this treatment in isolating the generalisability of the representation learning algorithm from the that of the linear head, we also define the *linear head bias*. It is the difference between the OOD test accuracy evaluated by OOD linear head and that evaluated by the ID linear head, i.e.

$$b = \text{acc}_o(f, c_o) - \text{acc}_o(f, c_i). \quad (\text{A.1})$$

In a related work, Taori et al. (2020) proposed to evaluate the *effective robustness* of OOD generalisation defined as $\rho = \text{acc}_i(f, c_i) - \text{acc}_o(f, c_i)$, which quantifies the drop in performance (e.g. accuracy) when evaluating the model on OOD test set vs. ID test set. A small ρ is desirable, as it indicates that the performance of the model is relatively insensitive to a distribution shift². However, we note that a simple decomposition of *effective robustness* (ρ) shows a hidden *linear head bias* (b) term

$$\begin{aligned} \underbrace{\text{acc}_i(f, c_i) - \text{acc}_o(f, c_i)}_{\text{effective robustness } \rho} &= \text{acc}_i(f, c_i) - \text{acc}_o(f, c_i) - \text{acc}_o(f, c_o) + \text{acc}_o(f, c_o) \\ &= \underbrace{\text{acc}_o(f, c_o) - \text{acc}_o(f, c_i)}_{\text{linear head bias } b} + \underbrace{\text{acc}_i(f, c_i) - \text{acc}_o(f, c_o)}_{\text{shift sensitivity } s}. \end{aligned} \quad (\text{A.2})$$

Thus, we remove the effect of the linear head bias by subtracting b from ρ and reporting the last term in eq. (A.2). We refer to this as *shift sensitivity*: $s = \rho - b$. Alternatively, it is the difference between the OOD accuracy using linear head trained on OOD data, and ID accuracy using linear head trained on ID data. Larger

²We negate the original definition of effective robustness from Taori et al. (2020) for ease of understanding.

s marks higher sensitivity of f to distribution shift, which is, possibly, dangerous for the deployment of such models. In summary, for each experiment we report the following three metrics: OOD linear head accuracy $\text{acc}_o(f, c_o)$, linear head bias b and shift sensitivity s .

A.3 Experimental results

We perform a hyperparameter search on learning rate, scheduler, optimiser, representation size, etc. for each model. We use the standard SSL augmentations proposed in He et al. (2020a); Chen et al. (2020c) for all models to ensure a fair comparison. See appendix F.2 for details.

A.3.1 Synthetic distribution shift

In this section, we evaluate the performance of SL, SSL and AE algorithms on synthetic distribution shift tasks, utilising the MNIST-CIFAR dataset (Shah et al., 2020) and the CdSprites dataset (Shi et al., 2022b). All results are averaged over 5 random seeds.

MNIST-CIFAR

Finding: *Under this extreme distribution shift setting, SSL and AE significantly outperform SL. The OOD accuracy of SSL and AE can be notably improved by retraining the linear head on OOD data, however the OOD accuracy of SL remains low even with the OOD-trained linear head.*

The MNIST-CIFAR dataset consists of concatenations of images from two classes of MNIST and CIFAR-10. In each concatenated image, the classes of the two datasets are either correlated or uncorrelated depending on the split as discussed below (See fig. A.1, *MNIST-CIFAR* for an example):

- **ID train, test:** Correlation between MNIST and CIFAR-10 labels is one. Each image belongs to one of the two classes: 1) MNIST “0” and CIFAR-10 “automobile”, and 2) MNIST “1” and CIFAR-10 “plane” (Figure A.1, *top row*);

- **OOD train, test:** Zero correlation between MNIST and CIFAR-10 labels, images from the two classes are randomly paired (Figure A.1, *bottom row*).

Since the MNIST features are much simpler than the CIFAR features, a model trained on the ID train set can use MNIST only to predict the label, even though the CIFAR images are just as predictive (Shah et al., 2020). This results in poor performance when predicting the CIFAR label on the OOD test set, where there is no correlation between the MNIST and CIFAR labels.

We train a CNN backbone on the ID train set using the eight SL, SSL and AE algorithms listed in appendix A.2. At test time, we freeze the backbone and train two linear heads on ID train and OOD train set respectively, and evaluate their performance on the ID and OOD test set to compute 1) OOD linear head accuracy $\text{acc}_o(f, c_o)$, 2) shift sensitivity s and, 3) linear head bias b . See results in table A.1.

We observe that all models achieve near perfect performance when predicting the MNIST label on OOD test set, all with low shift sensitivity and small linear head bias. However, when predicting the labels of the more complex CIFAR images, unsupervised algorithms have a clear advantage over the supervised one: SSL achieves the highest OOD accuracy at 86.1%, followed by AE at 79.9% and SL at 51.5% (near random). The shift sensitivity s of the three objectives follow a similar trend, with SSL and AE scoring significantly lower than SL. This indicates that unsupervised representations are significantly less sensitive to distribution shift compared to those from SL, with the latter suffering a drop as large as 47.6%. Interestingly, the classifier head bias b for SSL and AE are relatively high (around 30%), and is very low for SL (0.8%), indicating that the representations learned from SL is intrinsically un-robust to distribution shift. That is, while there exist (linearly separable) CIFAR features in the representations of SSL and AE that can be extracted using a linear head trained on un-biased (OOD) data, these features are absent from the representations of SL.

Table A.1: Evaluations on the MNIST-CIFAR dataset. We report accuracy on MNIST and CIFAR trained using OOD linear head ($\text{acc}_o(f, c_o)$), linear head bias (b) and shift sensitivity (s).

Regime	Method	MNIST (%)			CIFAR (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b	$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	99.9 ($\pm 1e-2$)	0.0 ($\pm 1e-2$)	0.0 ($\pm 2e-3$)	81.1 ($\pm 1e+0$)	18.8 ($\pm 1e+0$)	30.2 ($\pm 1e+0$)
	VAE	99.8 ($\pm 8e-3$)	-0.1 ($\pm 9e-3$)	0.5 ($\pm 1e-4$)	79.7 ($\pm 4e+0$)	20.2 ($\pm 3e+0$)	29.2 ($\pm 6e+0$)
	IWAE	99.8 ($\pm 9e-3$)	0.0 ($\pm 4e-3$)	0.1 ($\pm 5e-3$)	80.8 ($\pm 2e+0$)	19.0 ($\pm 3e+0$)	30.0 ($\pm 4e+0$)
	β -VAE	99.8 ($\pm 2e-2$)	0.0 ($\pm 4e-2$)	-0.1 ($\pm 3e-2$)	78.0 ($\pm 3e+0$)	21.8 ($\pm 4e+0$)	28.0 ($\pm 4e+0$)
AE average		99.8 ($\pm 1e-2$)	0.0 ($\pm 1e-2$)	0.1 ($\pm 9e-3$)	79.9 ($\pm 3e+0$)	20.0 ($\pm 4e+0$)	29.3 ($\pm 4e+0$)
SSL	SimCLR	99.7 ($\pm 1e-2$)	0.2 ($\pm 1e-3$)	-0.2 ($\pm 3e-3$)	85.8 ($\pm 1e+0$)	14.1 ($\pm 2e+0$)	35.5 ($\pm 1e+0$)
	SimSiam	99.8 ($\pm 2e-1$)	0.1 ($\pm 2e-1$)	0.0 ($\pm 9e-2$)	87.8 ($\pm 2e+0$)	12.1 ($\pm 2e+0$)	35.6 ($\pm 4e+0$)
	BYOL	99.8 ($\pm 4e-2$)	0.0 ($\pm 1e-2$)	0.9 ($\pm 8e-3$)	84.8 ($\pm 9e-1$)	15.0 ($\pm 1e+0$)	33.2 ($\pm 1e+0$)
SSL average		99.8 ($\pm 8e-2$)	0.1 ($\pm 5e-2$)	0.2 ($\pm 3e-2$)	86.1 ($\pm 2e+0$)	13.7 ($\pm 2e+0$)	34.8 ($\pm 4e+0$)
SL	Supervised	97.7 ($\pm 9e-1$)	1.4 ($\pm 1e+0$)	-0.3 ($\pm 1e+0$)	51.5 ($\pm 1e+0$)	47.6 ($\pm 1e+0$)	0.8 ($\pm 9e-1$)

CdSprites

Finding: Similar to MNIST-CIFAR, under extreme distribution shift, SSL and AE are better than SL; when the shift is less extreme, SSL and SL achieve comparably strong OOD generalisation performance while AE’s performance is much weaker.

CdSprites is a colored variant of the popular dSprites dataset (Matthey et al., 2017), which consists of images of 2D sprites that are procedurally generated from multiple latent factors. The CdSprites dataset induces a spurious correlation between the color and shape of the sprites, by coloring the sprites conditioned on the shape following a controllable correlation coefficient r_{id} . See fig. A.1 for an example: when $r_{\text{id}} = 1$ color is completely dependent on shape (*top row*, oval-purple, heart-cyan, square-white), and when $r_{\text{id}} = 0$, color and shape are randomly matched (*bottom row*).

Shi et al. (2022b) observes that when r_{id} is high, SL model tend to use color only to predict the label while ignoring shape features due to the texture bias of CNN (Geirhos et al., 2018; Brendel and Bethge, 2019). First, we consider the setting of extreme distribution shift similar to MNIST-CIFAR by setting $r_{\text{id}} = 1$ in the ID train and test splits. In the OOD train and test splits, the correlation coefficient is set to zero to investigate how well the model learns both the shape and the color features. Table A.2 reports the three metrics of interest using the same evaluation protocol as before.

Similar to MNIST-CIFAR, we observe that all models achieve near perfect performance when predicting the simpler feature, i.e. color on the OOD test set. However, when predicting shape, the more complex feature on the OOD test set, SSL (and also AEs to a lesser extent) is far superior to SL. Additionally, the shift sensitivity of SSL (and AE to a lesser extent) are much smaller than SL, indicating that SSL/AE models are more robust to extreme distribution shift. The linear head bias also follows a similar trend as for MNIST-CIFAR, showing that representations learned using SL methods are inherently not robust to spurious correlations. This is not the case for SSL and AE algorithms where a large linear head bias shows that is the ID linear heads and not the representations that injects the bias.

Controllable distribution shift We extend this experiment to probe the performance of these algorithms under *varying* degrees of distribution shifts. We generate three versions of the CdSprites dataset with three different correlation coefficients $r_{\text{id}} \in \{0, 0.5, 1\}$ of the ID train set. As before, the correlation coefficient of the OOD split is set to zero. The rest of the experimental protocol stays the same. The OOD test accuracy and the shift sensitivity for varying r_{id} is plotted in fig. A.3 and a detailed breakdown of results is available in appendix F.1.

Figure A.3 shows that despite increasing distribution shift between the ID and OOD splits (with increasing r_{id}) the OOD performance of SSL and AE does not suffer. However, the OOD accuracy of SL plummets and its shift sensitivity explodes at $r_{\text{id}} = 1$. Interestingly, SSL maintains a high OOD test accuracy regardless of the level of distribution shift: when $r_{\text{id}} < 1$ its performance is on par with SL, and when the distribution shift becomes extreme with $r_{\text{id}} = 1$ it significantly outperforms SL both in terms of accuracy and shift sensitivity. In comparison, AE models' accuracy lingers around 50%, with increasingly higher shift sensitivity as r_{id} increases. However, under extreme distribution shift with $r_{\text{id}} = 1$ it still performs better than SL, with slightly higher OOD accuracy and lower shift sensitivity.

Table A.2: Evaluations on the CdSprites dataset with $r_{id} = 1.0$. We report accuracy for color and shape classifiers trained using OOD linear head ($\text{acc}_o(f, c_o)$), linear head bias (b) and shift sensitivity (s).

Regime	Method	Color classification (%)			Shape classification (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b	$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	100.0 ($\pm 0e+0$)	0.0 ($\pm 2e-3$)	0.3 ($\pm 5e-1$)	46.1 ($\pm 6e-1$)	53.9 ($\pm 6e-1$)	12.7 ($\pm 5e-1$)
	VAE	99.7 ($\pm 3e-1$)	0.3 ($\pm 3e-1$)	-0.3 ($\pm 3e-1$)	52.4 ($\pm 2e+0$)	47.6 ($\pm 2e+0$)	18.9 ($\pm 3e+0$)
	IWAE	100.0 ($\pm 0e+0$)	0.0 ($\pm 2e-3$)	0.4 ($\pm 5e-1$)	58.9 ($\pm 2e+0$)	41.1 ($\pm 2e+0$)	25.6 ($\pm 2e+0$)
AE average		99.9 ($\pm 9e-1$)	0.1 ($\pm 9e-2$)	0.1 ($\pm 4e-1$)	52.5 ($\pm 2e+0$)	47.5 ($\pm 2e+0$)	19.1 ($\pm 2e+0$)
SSL	SimCLR	100.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	0.0 ($\pm 1e-1$)	87.8 ($\pm 5e-1$)	12.2 ($\pm 5e-1$)	54.5 ($\pm 5e-1$)
	SimSiam	100.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	0.1 ($\pm 1e-1$)	69.2 ($\pm 2e+0$)	30.8 ($\pm 2e+0$)	35.6 ($\pm 2e+0$)
	BYOL	100.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	91.1 ($\pm 4e+0$)	8.9 ($\pm 4e+0$)	57.9 ($\pm 4e+0$)
SSL average		100.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	0.1 ($\pm 3e-2$)	82.7 ($\pm 2e+0$)	17.3 ($\pm 2e+0$)	49.3 ($\pm 4e+0$)
SL	Supervised	100.0 ($\pm 0e+0$)	0.0 ($\pm 0e+0$)	0.0 ($\pm 3e-2$)	44.0 ($\pm 7e-1$)	56.0 ($\pm 7e-1$)	10.7 ($\pm 7e-1$)

A.3.2 Real-world distribution shift

In this section we investigate the performance of different objectives on real-world distribution shift tasks. We use two datasets from WILDS (Koh et al., 2021): 1) Camelyon17, which contains tissue scans acquired from different hospitals, and the task is to determine if a given patch contains breast cancer tissue; and 2) FMoW, which features satellite images of landscapes on five different continents, with the classification target as the type of infrastructure. See examples in Figure A.1. Following the guidelines from WILDS benchmark, we perform 10 random seed runs for all Camelyon17 experiment and 3 random seed runs for FMoW. The error margin in Figure A.3 represent standard deviation.

Original WILDS Datasets

Findings: *SL is significantly more sensitive to distribution shift than SSL and AE; representations from SSL obtain higher OOD accuracy than SL on Camelyon17 but lower on FMoW. AE is consistently the least sensitive to distribution shift though it has the lowest accuracy. The performance of all models significantly improves by retraining the linear head on a small amount of OOD data.*

The original Camelyon17 and FMoW dataset from WILDS benchmark both contains the following three splits: ID train, OOD validation and OOD test. We further create five splits specified as follows:

A. How Robust is Unsupervised Representation Learning to Distribution Shift?

- **ID train, test:** Contains 90% and 10% of the original ID train split, respectively;
- **OOD train, test:** Contains 10% and 90% of the original OOD test split, respectively;
- **OOD validation:** Same as the original OOD validation split.

Following WILDS, we use OOD validation set to perform early stopping and choose hyperparameters; we also use DenseNet-121 (Huang et al., 2017) as the backbone for all models. We follow similar evaluation protocol as previous experiments, and in addition adopt 10-fold cross-validation for the OOD train and test set. See results in Tables A.3 and A.4, where following WILDS, we report performance on Camelyon17 using standard average accuracy and on FMoW using worst-group accuracy.

One immediate observation is that in contrast to our previous experiments on synthetic datasets, SL’s OOD accuracy is much higher in comparison on realistic distribution shift tasks: it is the best performing model on FMoW with 35.6% worst-group accuracy on OOD test set; its OOD accuracy is the lowest on Camelyon17, however it is only 3% worse than the highest accuracy achieved by SSL (89.8%). This highlights the need to study realistic datasets along with synthetic ones. Nonetheless, we find that SSL is still the best performing method on Camelyon17 and achieves competitive performance on FMoW with accuracy 29.6% — despite learning without labels! AE has much lower OOD accuracy on FMoW compared to the other two methods: we believe this is due to its reconstruction-based objective wasting modelling capacity on high frequency details, a phenomenon frequently observed in prior work (Bao et al., 2021; Ramesh et al., 2021). Note that the standard deviation for all three methods are quite high for Camelyon17: this is a known property of the dataset and similar pattern is observed across most methods on WILDS benchmark (Koh et al., 2021).

Table A.3: Evaluations on test set of Camelyon17, all metrics computed using average accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	84.4 ($\pm 2e+0$)	-0.6 ($\pm 1e+0$)	12.7 ($\pm 2e+0$)
	VAE	88.1 ($\pm 2e+0$)	0.5 ($\pm 2e+0$)	39.0 ($\pm 2e+0$)
	IWAE	88.1 ($\pm 1e+0$)	-0.9 ($\pm 3e+0$)	39.1 ($\pm 4e+0$)
	β -VAE	87.1 ($\pm 4e+0$)	0.2 ($\pm 4e+0$)	36.0 ($\pm 5e+0$)
AE average		86.9 ($\pm 2e+0$)	-0.2 ($\pm 3e+0$)	31.7 ($\pm 3e+0$)
SSL	SimCLR	92.7 ($\pm 2e+0$)	0.4 ($\pm 1e+0$)	8.3 ($\pm 1e+0$)
	SimSiam	86.7 ($\pm 1e+0$)	3.1 ($\pm 1e+0$)	7.9 ($\pm 3e+0$)
	BYOL	89.9 ($\pm 1e+0$)	1.4 ($\pm 1e+0$)	10.3 ($\pm 2e+0$)
SSL average		89.8 ($\pm 1e+0$)	1.6 ($\pm 1e+0$)	8.8 ($\pm 2e+0$)
SL	Supervised	86.8 ($\pm 2e+0$)	3.5 ($\pm 1e+0$)	7.4 ($\pm 3e+0$)

Table A.4: Evaluations on test set of FMoW, all metrics computed using worst-group accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	26.9 ($\pm 9e-3$)	6.4 ($\pm 6e-3$)	5.8 ($\pm 1e-2$)
	VAE	21.7 ($\pm 6e-3$)	4.7 ($\pm 4e-3$)	8.0 ($\pm 2e-2$)
	IWAE	20.9 ($\pm 2e-2$)	5.5 ($\pm 1e-2$)	7.8 ($\pm 1e-2$)
	β -VAE	21.7 ($\pm 5e-3$)	3.4 ($\pm 6e-3$)	7.6 ($\pm 8e-3$)
AE average		22.8 ($\pm 3e-2$)	5.0 ($\pm 7e-3$)	7.3 ($\pm 1e-2$)
SSL	SimCLR	29.9 ($\pm 6e-3$)	10.7 ($\pm 6e-3$)	7.6 ($\pm 7e-3$)
	SimSiam	27.8 ($\pm 2e-2$)	4.6 ($\pm 1e-2$)	6.3 ($\pm 2e-2$)
	BYOL	31.3 ($\pm 1e-2$)	12.1 ($\pm 7e-3$)	7.9 ($\pm 1e-2$)
SSL average		29.6 ($\pm 2e-2$)	9.1 ($\pm 9e-3$)	7.3 ($\pm 1e-2$)
SL	Supervised	35.6 ($\pm 7e-3$)	37.7 ($\pm 4e-2$)	6.9 ($\pm 9e-3$)

In terms of shift sensitivity, unsupervised objectives including SSL and AE consistently outperforms SL — this stands out the most on FMoW, where the shift sensitivity of SSL and AE are 9.1% and 5.0% respectively, while SL is as high as 37.7%. This observation further validates our previous finding on synthetic datasets, that SSL and AE’s ID accuracy is a relatively reliable indication of their generalisation performance, while SL can undergo a huge performance drop under distribution shift, which can be dangerous for the deployment of such models. We highlight that, in sensitive application domains, a low shift sensitivity is an important criterion as it implies that the model’s performance will remain consistent when the distribution shifts. Another interesting observation here is that for all objectives on both datasets, the classifier bias b is consistently high. This indicates the bias of the linear classification head plays a significant role even for real world distribution shifts, and that it is possible to mitigate this effect by training the linear head using a small amount of OOD data (in this case 10% of the original OOD test set).

WILDS Datasets with Controllable Shift

Findings: *SL’s OOD accuracy drops as more the distribution shift becomes more challenging, with SSL being the best performing model when the distribution shift is the most extreme. The shift sensitivity of SSL and AE are consistently lower than SL regardless of the level of shift.*

To examine models’ generalisation performance under different levels of distribution shift, we create versions of these realistic datasets with *controllable shifts*, which

A. How Robust is Unsupervised Representation Learning to Distribution Shift?

Table A.5: Evaluations on test set of Camelyon17-C with $r_{\text{id}} = 0.5$, all metrics computed using average accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	80.4 ($\pm 3e+0$)	6.0 ($\pm 2e+0$)	19.0 ($\pm 4e+0$)
	VAE	88.6 ($\pm 2e+0$)	-0.5 ($\pm 1e+0$)	17.8 ($\pm 6e+0$)
	IWAE	87.8 ($\pm 1e+0$)	-0.2 ($\pm 1e+0$)	26.4 ($\pm 6e+0$)
	β -VAE	88.5 ($\pm 2e+0$)	0.1 ($\pm 9e-1$)	19.7 ($\pm 6e+0$)
AE average		86.3 ($\pm 2e+0$)	1.4 ($\pm 1e+0$)	20.7 ($\pm 5e+0$)
SSL	SimCLR	84.5 ($\pm 2e+0$)	8.0 ($\pm 1e+0$)	6.6 ($\pm 2e+0$)
	SimSiam	86.1 ($\pm 2e+0$)	5.7 ($\pm 1e+0$)	8.3 ($\pm 4e+0$)
	BYOL	86.4 ($\pm 2e+0$)	4.5 ($\pm 2e+0$)	8.8 ($\pm 4e+0$)
SSL average		85.7 ($\pm 2e+0$)	6.1 ($\pm 2e+0$)	7.9 ($\pm 3e+0$)
SL	Supervised	81.5 ($\pm 5e+0$)	13.2 ($\pm 3e+0$)	3.4 ($\pm 4e+0$)

Table A.7: Evaluations on test set of Camelyon17-C with $r_{\text{id}} = 1$, all metrics computed using average accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	75.7 ($\pm 5e+0$)	7.3 ($\pm 2e+0$)	35.1 ($\pm 4e+0$)
	VAE	86.0 ($\pm 3e+0$)	2.7 ($\pm 1e+0$)	12.4 ($\pm 4e+0$)
	IWAE	86.1 ($\pm 1e+0$)	2.6 ($\pm 7e-1$)	9.1 ($\pm 3e+0$)
	β -VAE	84.7 ($\pm 2e+0$)	3.8 ($\pm 1e+0$)	15.5 ($\pm 4e+0$)
AE average		83.1 ($\pm 3e+0$)	4.1 ($\pm 1e+0$)	18.0 ($\pm 4e+0$)
SSL	SimCLR	85.8 ($\pm 8e+1$)	2.8 ($\pm 4e-1$)	6.2 ($\pm 2e+0$)
	SimSiam	82.1 ($\pm 1e+0$)	6.0 ($\pm 7e-1$)	8.3 ($\pm 4e+0$)
	BYOL	74.8 ($\pm 5e+0$)	10.3 ($\pm 2e+0$)	-2.2 ($\pm 4e+0$)
SSL average		80.9 ($\pm 2e+0$)	6.3 ($\pm 1e+0$)	4.1 ($\pm 3e+0$)
SL	Supervised	73.4 ($\pm 6e+0$)	10.7 ($\pm 3e+0$)	5.9 ($\pm 8e+0$)

Table A.6: Evaluations on test set of FMoW-C with $r_{\text{id}} = 0.5$, all metrics computed using worst-group accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	23.4 ($\pm 1e+0$)	8.6 ($\pm 6e-1$)	4.2 ($\pm 8e-1$)
	VAE	18.7 ($\pm 1e+0$)	7.7 ($\pm 6e-1$)	1.5 ($\pm 6e-1$)
	IWAE	18.5 ($\pm 2e+0$)	7.3 ($\pm 1e+0$)	2.2 ($\pm 1e+0$)
	β -VAE	21.4 ($\pm 3e-1$)	4.0 ($\pm 4e-1$)	3.9 ($\pm 7e-1$)
AE average		20.5 ($\pm 2e+0$)	6.9 ($\pm 8e-1$)	3.0 ($\pm 9e-1$)
SSL	SimCLR	29.5 ($\pm 9e-1$)	9.2 ($\pm 6e-1$)	6.9 ($\pm 7e-1$)
	SimSiam	27.9 ($\pm 2e+0$)	7.5 ($\pm 1e+0$)	4.6 ($\pm 1e+0$)
	BYOL	32.6 ($\pm 3e+0$)	7.9 ($\pm 2e+0$)	8.5 ($\pm 2e+0$)
SSL average		30.0 ($\pm 2e+0$)	8.2 ($\pm 1e+0$)	6.7 ($\pm 1e+0$)
SL	Supervised	32.3 ($\pm 3e+0$)	25.1 ($\pm 3e+0$)	6.0 ($\pm 2e+0$)

Table A.8: Evaluations on test set of FMoW-C with $r_{\text{id}} = 1$, all metrics computed using worst-group accuracy.

Regime	Method	Metrics (%)		
		$\text{acc}_o(f, c_o) \uparrow$	$s \downarrow$	b
AE	AE	22.4 ($\pm 1e+0$)	10.0 ($\pm 6e-1$)	3.8 ($\pm 7e-1$)
	VAE	16.6 ($\pm 9e-1$)	8.6 ($\pm 1e+0$)	2.8 ($\pm 8e-1$)
	IWAE	17.2 ($\pm 5e-1$)	8.7 ($\pm 6e-1$)	3.8 ($\pm 5e-1$)
	β -VAE	16.7 ($\pm 3e-1$)	7.9 ($\pm 4e-1$)	3.2 ($\pm 4e-1$)
AE average		18.3 ($\pm 3e+0$)	8.8 ($\pm 7e-1$)	3.4 ($\pm 6e-1$)
SSL	SimCLR	26.3 ($\pm 1e+0$)	10.6 ($\pm 8e-1$)	7.2 ($\pm 1e+0$)
	SimSiam	27.1 ($\pm 5e-1$)	6.3 ($\pm 7e-1$)	7.8 ($\pm 3e-1$)
	BYOL	27.4 ($\pm 2e+0$)	11.1 ($\pm 1e+0$)	6.5 ($\pm 2e+0$)
SSL average		26.9 ($\pm 6e-1$)	9.3 ($\pm 1e+0$)	7.2 ($\pm 1e+0$)
SL	Supervised	25.7 ($\pm 5e-1$)	12.3 ($\pm 2e+0$)	5.3 ($\pm 1e+0$)

we name Camelyon17-CS and FMoW-CS. Specifically, we subsample the ID train set of these datasets to artificially create spurious correlation between the domain and label. For instance, given dataset with domain A, B and label 0, 1, to create a version of the dataset where the spurious correlation is 1 we would sample only examples with label 0 from domain A and label 1 from domain B. See Appendix F.3 for further details.

Similar to CdSprites, we create three versions of both of these datasets with the spurious correlation coefficient $r_{\text{id}} \in \{0, 0.5, 1\}$ in ID (train and test) sets. The OOD train, test and validation set remains unchanged³. Using identical experimental setup as in appendix A.3.2, we plot the trend of performance with increasing r for Camelyon17-CS and FMoW-CS in Figure A.3 with detailed numerical breakdown in Tables A.5 to A.8.

³Note that even when $r_{\text{id}} = 0$, distribution shift between the ID and OOD splits exists, as the spurious correlation is not the only source of the distribution shift.

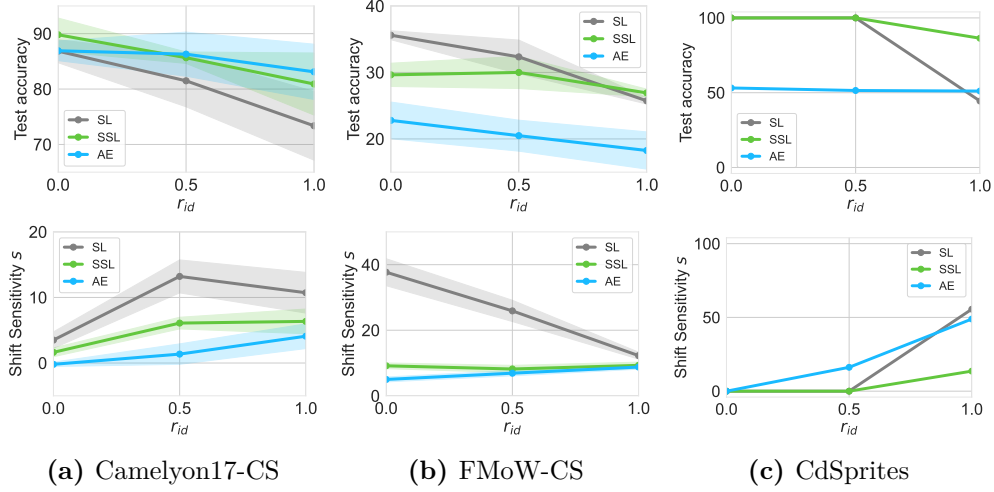


Figure A.3: Evaluations on Camelyon17-CS, FMoW-CS, and CdSprites with $r_{id} \in \{0, 0.5, 1.0\}$. We report OOD test accuracy using OOD-trained linear head ($\text{acc}_o(f, c_o)$) and shift sensitivity (s). Blue lines are results averaged over AE models, green lines are SSL models and grey is SL.

For both datasets, the OOD test accuracy of all models drop as the spurious correlation r_{id} increases (*top row* in Figure A.3). However, this drop is the far more obvious in SL than in SSL and AE: when $r_{id} = 1$, SL’s accuracy is 10% lower than SSL on Camelyon17 and 2% lower on FMoW — a significant drop from its original 3% lag on Camelyon17 and 5% lead on FMoW. This demonstrates that SL is less capable of dealing with more challenging distribution shift settings compared to SSL and AE. In terms of shift sensitivity (*bottom row*, fig. A.3), SL’s remains the highest regardless of r_{id} ; curiously, we see a decrease in SL’s shift sensitivity as r_{id} increases in FMoW-CS, however this has more to do with the ID test set accuracy decreasing due to the subsampling of the dataset.

A.4 Related Work

While we are the first to systematically evaluate the OOD generalisation performance of unsupervised learning algorithms, there are other insightful work that considers the robustness to distribution shift of other existing, non-specialised methods/techniques. For instance, Liu et al. (2022c) studies the impact of different pre-training set-ups to distribution shift robustness, including dataset, objective and data augmentation. Ghosal et al. (2022) focuses on architecture, and found that Vision Transformers are more robust to spurious correlations than ConvNets

when using larger models and are given more training data; further, Liu et al. (2021) found that SSL is more robust to data imbalance. Azizi et al. (2022) also performed extensive studies on the generalisation performance of SSL algorithms on medical data. Interestingly, Robinson et al. (2021) also investigates the robustness of contrastive-SSL methods against extreme spurious correlation (i.e. simplicity bias). However, their work did not consider the linear head bias found in (Kirichenko et al., 2022; Kang et al., 2020) and led to opposing conclusions. In contrast, our work investigates the distribution shift performance of unsupervised algorithms, with experiments on both synthetic and realistic settings that go beyond the data imbalance regime. By isolating the linear head bias in our experiments, we find that unsupervised, especially SSL-learned representations, achieves similar if not better generalisation performance than SL under a wide range of distribution shift settings. See Appendix F.4 for a more detailed discussion on distribution shift problems.

A.5 Conclusion and future work

In this paper, we investigate the robustness of both unsupervised (AE, SSL) and supervised (SL) objectives for distribution shift. Through extensive and principled experiments on both synthetic and realistic distribution shift tasks, we find unsupervised representation learning algorithms to consistently outperform SL when the distribution shift is extreme. In addition, we see that SSL’s OOD accuracy is comparable, if not better to SL in all experiments. This is particularly crucial, as most work studying distribution shift for images are developed in the SL regime. We hope that these results inspire more future work on unsupervised/semi-supervised representation learning methods for OOD generalisation. Another important finding is that unsupervised models’ performance remains relatively stable under distribution shift. This is especially crucial for the real-world application of these machine learning systems, as this indicates that the ID performance of SSL/AE algorithms are a more reliable indicator of how they would perform in different environments at deployment, while that of SL is not. It is also worth noting that while models trained with AE objectives are consistently the least sensitive to distribution shift on realistic datasets, their OOD performance can be low especially when presented with complex

data (such as FMoW). This is consistent to the observation in prior work that these models can waste modelling capacity on high frequency details, and suggests that one should be careful about employing AE algorithms on large scale, complex tasks. Finally, a key contribution of this work is establishing the existence of linear head bias even for realistic distribution shift problems. We believe that using an OOD-trained linear head is necessary to be able to make comparisons between various algorithms irrespective of the final downstream task, and on the other hand, more efforts in the field of distribution shift could be devoted into re-balancing the linear layer.

B

Appendix: Variational Mixture-of-Experts Autoencoders for Multimodal Deep Generative Models

B.1 Tighter lower bound

With the MoE joint posterior, an alternative way of extending the $\mathcal{L}_{\text{IWAE}}$ in Equation (3.2) to M -modalities is by employing stratified sampling (Robert and Casella, 2013)—to take K samples from the joint posterior, we first sample a modality m , then take $L = K/M$ samples from the corresponding marginal variational posterior $q_{\phi_m}(\mathbf{z} \mid \mathbf{x}_m)$, and repeat the process M times. Formally,

$$\underline{\mathcal{L}}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}) = \mathbb{E}_{\substack{\mathbf{z}_1^{1:L} \sim q_{\phi_1}(\mathbf{z} \mid \mathbf{x}_1) \\ \vdots \\ \mathbf{z}_M^{1:L} \sim q_{\phi_M}(\mathbf{z} \mid \mathbf{x}_M)}} \left[\log \frac{1}{M} \sum_{m=1}^M \frac{1}{L} \sum_{l=1}^L \frac{p_{\theta}(\mathbf{z}_m^l, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z}_m^l \mid \mathbf{x}_{1:M})} \right]. \quad (\text{B.1})$$

Applying Jensen’s inequality, we see that Equation (B.1) is a tighter lower bound than Equation (3.3); that is, $\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}) \leq \underline{\mathcal{L}}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}) \leq \log p_{\theta}(\mathbf{x}_{1:M})$; however, it can adversely affect the ability to perform cross-modal generation. To see this, consider the form of the gradient estimator of this objective with respect to the

variational posterior parameters Φ (Burda et al., 2015; Cremer et al., 2017),

$$\nabla_{\Phi} \mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M}) = \mathbb{E}_{\epsilon^{1:K} \sim p(\epsilon)} \left[\sum_k \bar{w}^k \nabla_{\Phi} \log w^k \right], \quad (\text{B.2})$$

where $K = ML$, $\mathbf{z}^k = g(\epsilon^k, \Phi)$ (reparameterised), $w^k = \frac{p_{\theta}(\mathbf{z}^k, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z}^k | \mathbf{x}_{1:M})}$, and $\bar{w}^k = \frac{w^k}{\sum_{j=1}^K w^j}$, indicating that the gradient weights samples by their relative importance $\bar{w}^k$. Two different samples $\mathbf{z}^{k_1}$, and $\mathbf{z}^{k_2}$ coming from different modalities (encoders) can have their gradients weighed differently from one another, leading to situations where the joint variational posterior collapses to one of the experts in the mixture. Figure B.1 shows empirical evidence of this happening, comparing performance under Equations (3.3) and (B.1) for the same data.

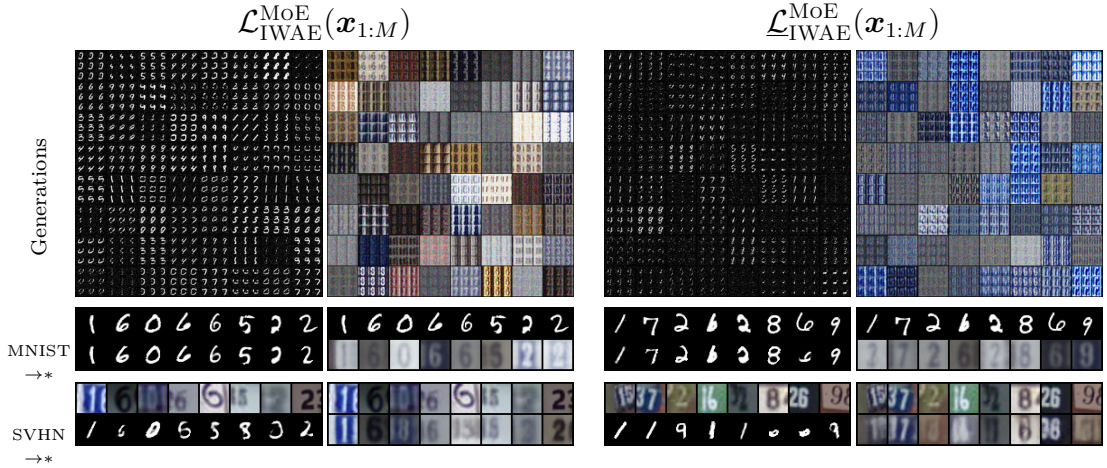


Figure B.1: Comparing performance for MNIST-SVHN between the $\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$ and $\hat{\mathcal{L}}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$ objectives.

Compared to the original objective (LHS), the cross generation of $\hat{\mathcal{L}}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$ struggles to match digits between the modalities, especially in the SVHN $\rightarrow$ MNIST case. Coherence for joint generation is also worse, with no recognisable matching-up of digits between the two modalities. Similar to Table 3.2, we evaluate the coherence of cross and joint generation by computing the accuracy of digit predictions between the two modalities in Table B.1. Note the drop in performance for the $\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$ objective reflecting the qualitative results.

Objective	Joint	Cross (M → S)	Cross (S → M)
$\mathcal{L}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$	42.1	86.4	69.1
$\underline{\mathcal{L}}_{\text{IWAE}}^{\text{MoE}}(\mathbf{x}_{1:M})$	24.2	74.6	15.6

Table B.1: Probability of digit matching (%) for joint and cross generation.

B.2 Multimodal Importance-Sampled ELBO

Consider the ELBO, with the basic MoE variational posterior,

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}_{1:M}) = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{z} \sim q_{\phi_m}(\mathbf{z} | \mathbf{x}_m)} \left[\log \frac{p_{\Theta}(\mathbf{z}, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M})} \right].$$

The term corresponding to a particular modality i is given as

$$\begin{aligned} & \mathbb{E}_{\mathbf{z}_i \sim q_{\phi_i}(\mathbf{z} | \mathbf{x}_i)} \left[\log \frac{p_{\theta}(\mathbf{z}_i, \mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_M)}{q_{\phi}(\mathbf{z}_i | \mathbf{x}_{1:M})} \right] \\ &= \mathbb{E}_{\mathbf{z}_i \sim q_{\phi_i}(\mathbf{z} | \mathbf{x}_i)} \left[\log \frac{p_{\theta}(\mathbf{z}_i, \mathbf{x}_i)}{q_{\phi}(\mathbf{z}_i | \mathbf{x}_{1:M})} \right] + \sum_{\substack{j=1 \\ j \neq i}}^M \mathbb{E}_{\mathbf{z}_i \sim q_{\phi_i}(\mathbf{z} | \mathbf{x}_i)} \left[\log p_{\theta_j}(\mathbf{x}_j | \mathbf{z}_i) \right] \\ &= \mathbb{E}_{\mathbf{z}_i \sim q_{\phi_i}(\mathbf{z} | \mathbf{x}_i)} \left[\log \frac{p_{\theta}(\mathbf{z}_i, \mathbf{x}_i)}{q_{\phi}(\mathbf{z}_i | \mathbf{x}_{1:M})} \right] + \underbrace{\sum_{\substack{j=1 \\ j \neq i}}^M \mathbb{E}_{\mathbf{z}_j \sim \bar{q}_{\phi_j}(\mathbf{z} | \mathbf{x}_j)} \left[\frac{q_{\phi_i}(\mathbf{z}_j | \mathbf{x}_i)}{\bar{q}_{\phi_j}(\mathbf{z}_j | \mathbf{x}_j)} \log p_{\theta_j}(\mathbf{x}_j | \mathbf{z}_j) \right]}_{A_j} \end{aligned}$$

where $\bar{q}$ does not propagate gradients (effectively issuing a `stop_gradient`), since all the likelihoods in the second term only have gradients with respect to $\phi_i, \theta_j, j \neq i$. Each term A_j can be seen as an importance-sampled estimate of $\mathbb{E}_{\mathbf{z}_i \sim q_{\phi_i}(\mathbf{z} | \mathbf{x}_i)} [\log p_{\theta_j}(\mathbf{x}_j | \mathbf{z}_i)]$ using that modality j 's own encoding distribution $q_{\phi_j}(\mathbf{z}_j | \mathbf{x}_j)$. This has two major benefits:

- i) it allows the total objective to be computed with just a single pass over each encoder and decoder to make computation linear in M . To see this is true, note that one can effectively precompute the likelihoods of modality j using samples from its own encoder, and simply weigh them by the appropriate ratio of variational posteriors where necessary.
- ii) estimating the likelihood of modality j using samples from modality i , where $i \neq j$, is a difficult ask—since observation $\mathbf{x}_i$ only carries partial information applicable for the reconstruction of $\mathbf{x}_j$. This multimodal importance sampling sidesteps the issue by using samples from same modality j , with appropriate

weighting. We would expect this to generally thus produce a lower-variance estimator, as it avoids potentially evaluating $\log p_{\theta_j}(\mathbf{x}_j \mid \mathbf{z}_i)$ for values $\mathbf{z}_i$ which yield very low likelihoods on $\mathbf{x}_j$. Moreover, since the denominators of the importance ratios cannot propagate gradients, the only way to improve the estimate is by maximising the density of samples from modality j in the variational posterior for i as $q_{\phi_i}(\mathbf{z}_j \mid \mathbf{x}_i)$, bringing the different components closer to each other.

B.3 The DReG Estimator

The standard gradient estimator of IWAE can have undesirably high variance (Roeder et al., 2017; Rainforth et al., 2018). To see this, we can expand Equation (B.2) as:

$$\nabla_{\Phi} \mathbb{E}_{z^{1:K}} \left[\log \left(\frac{1}{K} \sum_{k=1}^K w^k \right) \right] = \mathbb{E}_{\epsilon^{1:K}} \left[\sum_{k=1}^K \frac{w^k}{\sum_{j=1}^K w^j} \left(-\frac{\partial}{\partial \Phi} \log q_{\phi}(\mathbf{z}^k \mid \mathbf{x}) + \frac{\partial \log w^k}{\partial \mathbf{z}^k} \frac{d\mathbf{z}^k}{d\Phi} \right) \right] \quad (\text{B.3})$$

Roeder et al. (2017) find that when $K > 1$, the first term within paranthesis in Equation (B.3) need not be zero even when the approximate posterior matches true posterior everywhere, which can contribute to significant variance in the gradient estimator. To alleviate this, Tucker et al. (2019) re-apply the reparametrisation trick on it, yielding a Doubly Reparametrised Gradient Estimator (DReG):

$$\nabla_{\Phi} E_{z^{1:K}} \left[\log \left(\frac{1}{K} \sum_{k=1}^K w^k \right) \right] = E_{\epsilon^{1:K}} \left[\sum_{k=1}^K \left(\frac{w^k}{\sum_{j=1}^K w^j} \right)^2 \frac{\partial \log w^k}{\partial \mathbf{z}^k} \frac{d\mathbf{z}^k}{d\Phi} \right] \quad (\text{B.4})$$

We implement the estimator specified in Equation (B.4) when performing gradient updates for any experiment involving the IWAE objective.

B.4 Higher entropy of IWAE variational posteriors

To see why the variational posteriors estimated by IWAE tend to have higher entropy than those by ELBO, it is beneficial to consider the objectives from a different perspective.

First, let’s take a look at ELBO: maximising the standard ELBO indirectly minimises the KL between the variational and true posteriors, since

$$\log p_\theta(\mathbf{x}_{1:M}) = \mathcal{L}_{\text{ELBO}}(\mathbf{x}_{1:M}) + \text{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}_{1:M}) \parallel p_\theta(\mathbf{z} \mid \mathbf{x}_{1:M})).$$

Maximising the IWAE objective however, indirectly minimises the KL between *implicit* posteriors (Le et al., 2018) as

$$\log p_\theta(\mathbf{x}_{1:M}) = \mathcal{L}_{\text{IWAE}}(\mathbf{x}_{1:M}) + \text{KL}(q_{\Phi^{IS}}(\mathbf{z} \mid \mathbf{x}_{1:M}) \parallel p_{\Theta^{IS}}(\mathbf{z} \mid \mathbf{x}_{1:M})),$$

where

$$p_{\Theta^{IS}}(\mathbf{z} \mid \mathbf{x}_{1:M}) = \frac{1}{K} \sum_k \frac{q_{\Phi^{IS}}(\mathbf{z} \mid \mathbf{x}_{1:M})}{q_\phi(\mathbf{z}^k \mid \mathbf{x}_{1:M})} p_\theta(\mathbf{z}^k \mid \mathbf{x}_{1:M}),$$

$$q_{\Phi^{IS}}(\mathbf{z} \mid \mathbf{x}_{1:M}) = \frac{1}{K} \prod_k q_\phi(\mathbf{z}^k \mid \mathbf{x}_{1:M}),$$

leading to higher-entropy estimates of the variational posterior.

This suits the learning of multimodal data as each modality’s posterior attempts to explain *more* than just its own modality.

B.5 Qualitative results of MVAE implementation with MoE posterior

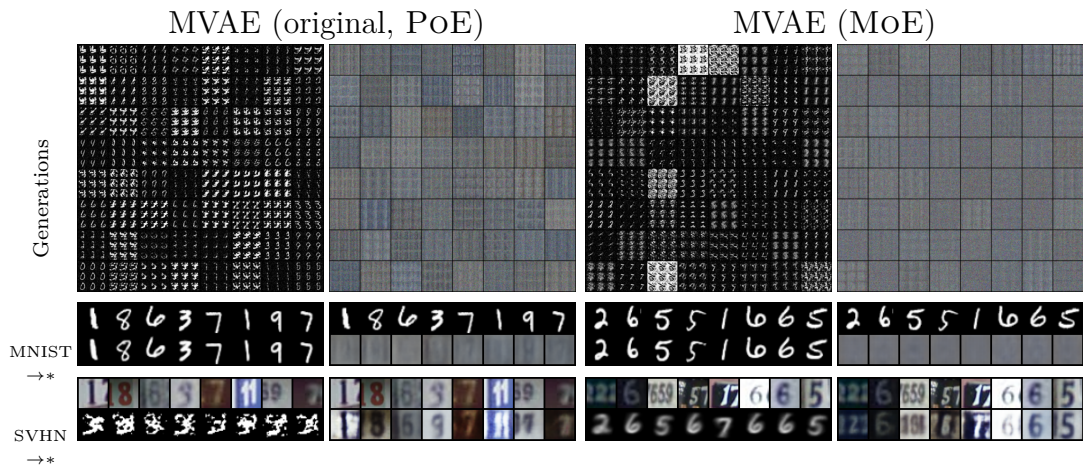


Figure B.2: A comparison of PoE vs. MoE in the MVAE codebase.

Here we show a qualitative evaluation in the MVAE codebase, minimally altered to use a MoE joint approximate posterior, with the results of the original PoE-MVAE

model as a comparison. We can see that MOE is able to generate recognisable MNIST digits from SVHN inputs (bottom row, column 3), while the original model fails completely at cross-modal generation. Although, do note that neither model performs well at coherence joint generation (top row).

B.6 Encoder and decoder architectures

Encoder	Decoder
Input $\in \mathbb{R}^{1 \times 28 \times 28}$	Input $\in \mathbb{R}^L$
FC. 400 ReLU	FC. 400 ReLU
FC. L , FC. L	FC. $1 \times 28 \times 28$ Sigmoid

(a) MNIST dataset.

Encoder
Input $\in \mathbb{R}^{1 \times 28 \times 28}$
4x4 conv. 32 stride 2 pad 1 & ReLU
4x4 conv. 64 stride 2 pad 1 & ReLU
4x4 conv. 128 stride 2 pad 1 & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1 pad 0
Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. 128 stride 1 pad 0 & ReLU
4x4 upconv. 64 stride 2 pad 1 & ReLU
4x4 upconv. 32 stride 2 pad 1 & ReLU
4x4 upconv. 3 stride 2 pad 1 & Sigmoid

(b) SVHN dataset.

Encoder	Decoder
Input $\in \mathbb{R}^{2048}$	Input $\in \mathbb{R}^L$
FC. 1024 ELU	FC. 256 ELU
FC. 512 ELU	FC. 512 ELU
FC. 256 ELU	FC. 1024 ELU
FC. L , FC. L	FC. 2048

(c) CUB image dataset.

Encoder
Input $\in \mathbb{R}^{1590}$
Word Emb. 256
4x4 conv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 128 stride 2 pad 1 & BatchNorm2d & ReLU
1x4 conv. 256 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
1x4 conv. 512 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1 pad 0
Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. 512 stride 1 pad 0 & ReLU
1x4 upconv. 256 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
1x4 upconv. 128 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
4x4 upconv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 1 stride 2 pad 1 & ReLU
Word Emb. ^T 1590

(d) CUB-Language dataset.

Table B.2: Encoder and decoder architectures.

B.7 Qualitative results of MMVAE on CUB

In this section, we show some more qualitative results of our MMVAE model on

CUB Image-Caption dataset.

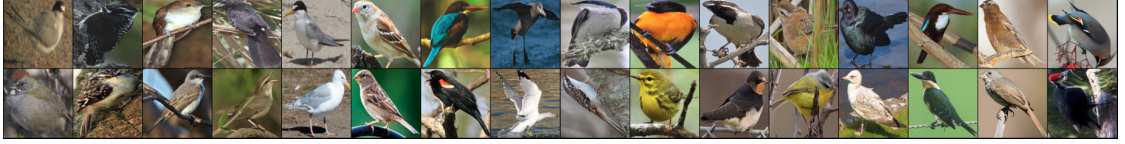


Figure B.3: Image reconstruction of MMVAE on CUB Image-Caption dataset. Top row: ground truth, bottom row: reconstruction.

[DATA] ==> a small bird with a white belly and a grey wing and a short beak. <eos> [RECON] ==> a small bird with a white belly and a brown , and a short beak. <eos>	[DATA] ==> bird has brown body feathers , brown breast feathers , and brown beak . <eos> [RECON] ==> bird has brown body feathers , brown breast feather , and black beak . <eos>
[DATA] ==> this bird is brown with white and has a long , pointy beak . <eos> [RECON] ==> this bird is black with white and has a long , pointy beak . <eos>	[DATA] ==> this is an orange bird with a black wing and head . <eos> [RECON] ==> this is a red bird with a wings and and beak . <eos>
[DATA] ==> this bird has wings that are grey and has a yellow tail . <eos> [RECON] ==> this bird has wings that are grey and has a yellow belly . <eos>	[DATA] ==> this bird is yellow with black on its head and has a long , pointy beak . <eos> [RECON] ==> this bird is yellow with grey on its head and has a long , pointy beak . <eos>
[DATA] ==> this bird has a white crown as well as a long black bill . <eos> [RECON] ==> this bird has a white crown , black primaries , and a white . <eos>	[DATA] ==> this bird has a speckled belly and breast with a short pointy bill . <eos> [RECON] ==> this bird has a speckled belly and breast with a short pointy bill . <eos>
[DATA] ==> this bird is white with brown and has a very short beak . <eos> [RECON] ==> this bird is white with belly and and a short beak . <eos>	[DATA] ==> this bird has black feathers and a thick black bill . <eos> [RECON] ==> this bird is completely black with a black orange bill . <eos>
[DATA] ==> this particular bird has a belly that is white and black . <eos> [RECON] ==> this particular bird has a belly that is black and white . <eos>	[DATA] ==> the bird has a grey belly and white coverts on the wingbars . <eos> [RECON] ==> this bird has a black crown and black with and black bill . <eos>

Figure B.4: Caption reconstruction of MMVAE on CUB Image-Caption dataset.



Figure B.5: Cross generation (a, b) and joint generation from prior samples (c) of MMVAE on CUB Image-Caption dataset.

B.8 Qualitative results of Wu and Goodman (2018a)’s MVAE on CUB

The qualitative results of MVAE on CUB Image-Caption dataset are as shown in Figure B.6 and fig. B.7. Note that similar to the MMVAE experiments, for the generation in the vision modality, we reconstruct the image features extracted from ResNet101 and perform nearest neighbour search to find the corresponding images.

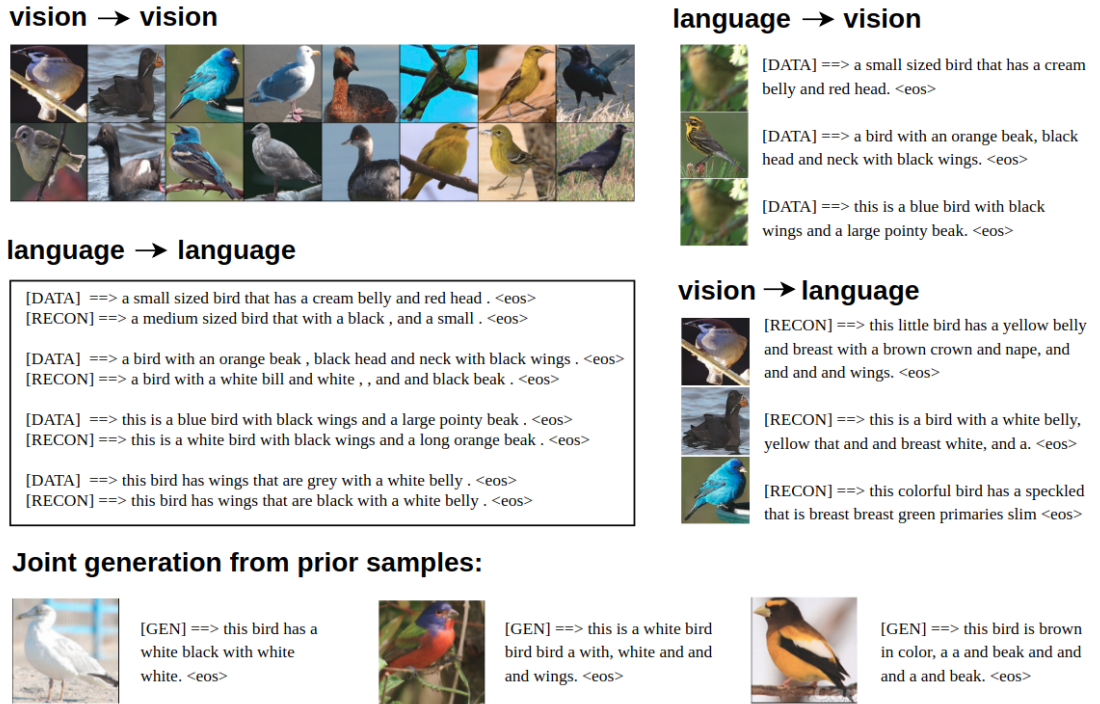


Figure B.6: Qualitative results of MVAE on CUB Image-Caption dataset, including reconstruction (vision → vision, language → language), cross generation (vision → language, language → vision) and joint generation from prior samples.

C

Appendix: Relating by Contrasting: A Data-efficient Framework for Multimodal DGM

C.1 From pointwise mutual information to joint marginal likelihood

In this section, we show that for the purpose of utilising the relatedness between multimodal pairs, maximising the difference between *pointwise mutual information* between related points $\mathbf{x}, \mathbf{y}$ and unrelated points $\mathbf{x}, \mathbf{y}'$ is equivalent to maximising the difference between their log *joint marginal likelihoods*.

To see this, we can expand $I(\mathbf{x}, \mathbf{y}) - I(\mathbf{x}, \mathbf{y}')$ as

$$\begin{aligned} I(\mathbf{x}, \mathbf{y}) - I(\mathbf{x}, \mathbf{y}') &= [\log p_{\Theta}(\mathbf{x}, \mathbf{y}) - \log p_{\Theta}(\mathbf{x}) - \log p_{\Theta}(\mathbf{y})] - [\log p_{\Theta}(\mathbf{x}, \mathbf{y}') - \log p_{\Theta}(\mathbf{x}) - \log p_{\Theta}(\mathbf{y}')] \\ &= \underbrace{\log p_{\Theta}(\mathbf{x}, \mathbf{y}) - \log p_{\Theta}(\mathbf{x}, \mathbf{y}')}_{\textcircled{1}} + \underbrace{\log p_{\Theta}(\mathbf{y}') - \log p_{\Theta}(\mathbf{y})}_{\textcircled{2}}. \end{aligned} \quad (\text{C.1})$$

In Equation (C.1) the PMI difference is decomposed as two terms: $\textcircled{1}$ the difference between joint marginal likelihoods and $\textcircled{2}$ the difference between marginal likelihoods of different instances of $\mathbf{y}$ -s. It is clear that since $\textcircled{2}$ involves only one modality and

only accounts for the difference in values between $\mathbf{y}$ and $\mathbf{y}'$, it is not relevant to the relatedness of $\mathbf{x}$, $\mathbf{y}$.

Therefore, for the purpose of utilising relatedness information, we only need to maximise $I(\mathbf{x}, \mathbf{y}) - I(\mathbf{x}, \mathbf{y}')$ through maximising term ①, i.e. the difference between joint marginal likelihoods.

C.2 Connection of final objective to Pointwise Mutual Information

Here, we show that minimising the objective in Equation (4.4) maximises the PMI between $\mathbf{x}$ and $\mathbf{y}$:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{y}) &= -\gamma \log p_{\Theta}(\mathbf{x}, \mathbf{y}) + \frac{1}{2} \left(\log \sum_{\mathbf{x}' \in X} p_{\Theta}(\mathbf{x}', \mathbf{y}) + \log \sum_{\mathbf{y}' \in Y} p_{\Theta}(\mathbf{x}, \mathbf{y}') \right) \\ &= -(\gamma - \frac{1}{2}) \log p_{\Theta}(\mathbf{x}, \mathbf{y}) - \frac{1}{2} \log \frac{p_{\Theta}(\mathbf{x}, \mathbf{y})}{\sum_{\mathbf{y}' \in Y} p_{\Theta}(\mathbf{x}, \mathbf{y}') \sum_{\mathbf{x}' \in X} p_{\Theta}(\mathbf{y}, \mathbf{x}')} \\ &\approx -(\gamma - \frac{1}{2}) \log p_{\Theta}(\mathbf{x}, \mathbf{y}) - \frac{1}{2} \log \underbrace{\frac{p_{\Theta}(\mathbf{x}, \mathbf{y})}{p_{\Theta}(\mathbf{x})p_{\Theta}(\mathbf{y})}}_{I(\mathbf{x}, \mathbf{y})} \end{aligned} \quad (\text{C.2})$$

We see in Equation (C.2) that minimising $\mathcal{L}$ can be decomposed to maximising both the joint marginal likelihood $p_{\Theta}(\mathbf{x}, \mathbf{y})$ and an **approximation** of PMI $I(\mathbf{x}, \mathbf{y})$. Note that since $\gamma > 1$, we can be sure that the joint marginal likelihood weighting $\gamma - \frac{1}{2}$ is non-negative.

C.3 Generalisation to $M > 2$ Modalities

In this section we show how the contrastive loss generalise to cases where number of modalities considered M is greater than 2.

Given observations from M modalities $D = \{X_1, X_2, \dots, X_m, \dots, X_M\}$, where X_m denotes unimodal dataset of modality m of size N_m , i.e. $X_m = \{\mathbf{x}_m^{(i)}\}_{i=1}^{N_m}$. Similar to Equation (4.2), we can write the asymmetrical contrastive loss for any observation $\mathbf{x}_m^{(i)}$ from modality m , where negative samples are taken for all $(M - 1)$ other modalities:

$$\mathcal{L}_C(\mathbf{x}_m^{(i)}, D_{\bar{m}}) = -\log p_{\Theta}(\mathbf{x}_{1:M}^{(i)}) + \log \sum_{\substack{d=1 \\ d \neq m}}^M \sum_{\substack{j=1 \\ j \neq i}}^N p_{\Theta}(\mathbf{x}_{1:(d-1),(d+1):M}^{(i)}, \mathbf{x}_d^{(j)}). \quad (\text{C.3})$$

We can therefore rewrite Equation (4.3) as:

$$\begin{aligned} \mathcal{L}_C(\mathbf{x}_{1:M}^{(i)}) &= \frac{1}{M} \sum_{m=1}^M \mathcal{L}_C(\mathbf{x}_m^{(i)}, D_{\bar{m}}) \\ &= -\log p_{\Theta}(\mathbf{x}_{1:M}^{(i)}) + \frac{1}{M} \sum_{m=1}^M \left(\log \sum_{\substack{d=1 \\ d \neq m}}^M \sum_{\substack{j=1 \\ j \neq i}}^N p_{\Theta}(\mathbf{x}_{1:(d-1),(d+1):M}^{(i)}, \mathbf{x}_d^{(j)}) \right), \end{aligned} \quad (\text{C.4})$$

$$(\text{C.5})$$

where N is the number of negative samples, all $\log p_{\Theta}(\mathbf{x}_{1:M})$ are approximated by the following joint ELBO for M modalities:

$$\log p_{\theta}(\mathbf{x}_{1:M}) \geq \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x}_{1:M})} \left[\log \frac{p_{\theta}(\mathbf{z}, \mathbf{x}_{1:M})}{q_{\phi}(\mathbf{z} | \mathbf{x}_{1:M})} \right] = \text{ELBO}(\mathbf{x}_{1:M}). \quad (\text{C.6})$$

While the above gives us the true generalisation of Equation (4.3), we note that the number of times where ELBO needs to be evaluated in Equation (C.5) is $\mathcal{O}(M^2N)$, making it difficult to implement this objective in practice, especially on more complicated datasets. We therefore propose a simplified version of the objective, where we estimate the second term of Equation (C.5) with N sets of random samples from all modalities. Specifically, we can precompute the following $M \times N$ random index matrix J :

$$J = \begin{bmatrix} j_{11} & j_{12} & \cdots & j_{1N} \\ j_{21} & j_{22} & \cdots & j_{2N} \\ \vdots & & & \vdots \\ j_{M1} & j_{M2} & \cdots & j_{MN} \end{bmatrix}, \quad (\text{C.7})$$

where each entry of J is a random integer taken from range $[1, N_m]$. We can then replace the second term of Equation (C.5) random samples selected by the indices in J , giving us

$$\mathcal{L}_C(\mathbf{x}_{1:M}^{(i)}) \approx -\log p_{\Theta}(\mathbf{x}_{1:M}^{(i)}) + \left(\log \sum_{n=1}^N p_{\Theta}(\mathbf{x}_1^{(J_{1n})}, \mathbf{x}_2^{(J_{2n})}, \dots, \mathbf{x}_M^{(J_{Mn})}) \right). \quad (\text{C.8})$$

The number of times ELBO needs to be computed is now $\mathcal{O}(N)$, and is no longer relevant to the number of modalities M .

We can now also generalise the final objective in Equation (4.4) to M modalities:

$$\mathcal{L}(\mathbf{x}_{1:M}^{(i)}) = -\gamma \text{ELBO}(\mathbf{x}_{1:M}^{(i)}) + \left(\log \sum_{n \in N} \text{ELBO}(\mathbf{x}_1^{(J_{1n})}, \mathbf{x}_2^{(J_{2n})}, \dots, \mathbf{x}_M^{(J_{Mn})}) \right). \quad (\text{C.9})$$

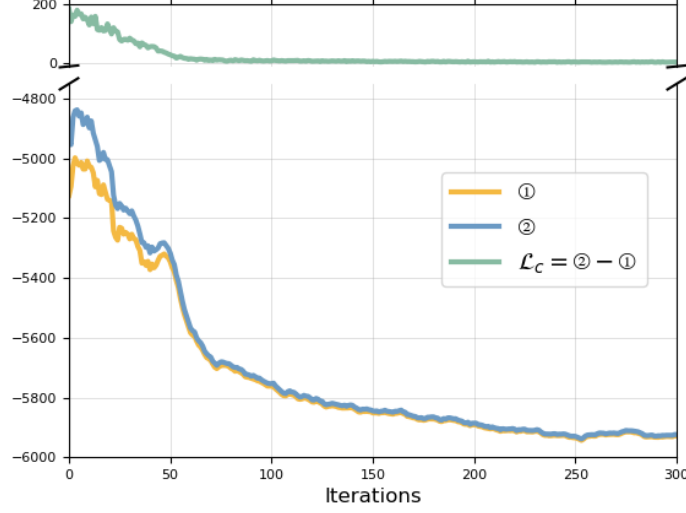


Figure C.1: First 300 iterations of training using contrastive loss $\mathcal{L}_C$ only.

C.4 The Ineffectiveness of training with $\mathcal{L}_C$ only

We demonstrate why training with the contrastive loss proposed in Equation (4.3) is ineffective, and why additional ELBO term is needed for the final objective. As we show in Figure C.1, when training with $\mathcal{L}_C$ only, while the contrastive loss (green) quickly drops to zero, both term ① and ② in Equation (4.3) also reduces drastically. This means the joint marginal likelihood of any generation $\log p_{\Theta}(\mathbf{x}, \mathbf{y})$ is small regardless the relatedness of $(\mathbf{x}, \mathbf{y})$. In comparison, we also plot the training curve for model trained on the final objective in Equation (4.4), which upweights term ① in Equation (4.3) by γ . We see in Figure C.2 that by setting $\gamma = 2$, the joint marginal likelihood (yellow and blue curve) improves during training, while $\mathcal{L}_C$ (green curve) gets minimised.

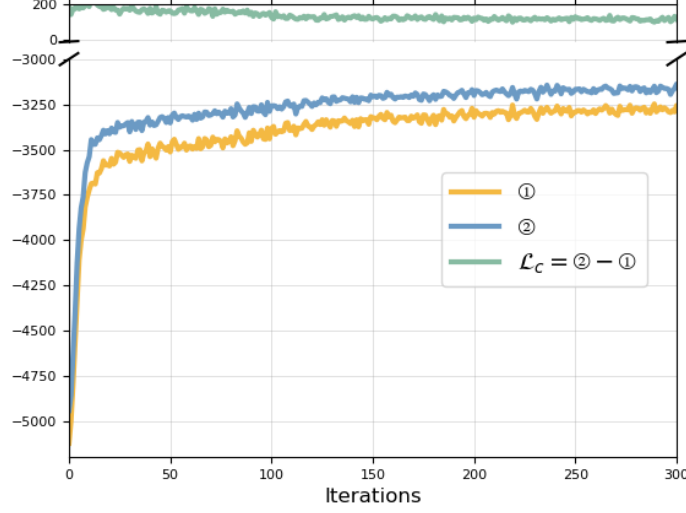


Figure C.2: First 300 iterations of training with final loss $\mathcal{L}$, where $\gamma = 2$.

C.5 Ablation study of γ

In Section 4.3, we specified that γ needs to be greater than 1 to offset the negative effect of minimising ELBO through term ② in Equation (4.4). Here, we study the effect of γ in details. Here we fix the value of $N = 5$ to ensure that γ solely controls the scaling between term ① and ② in our objective (Equation (4.4)).

Figure C.3 compares latent accuracy, cross coherence and joint coherence of MMVAE on MNIST-SVHN dataset trained on different values of γ . Note that here we only consider cases where $\gamma \geq 1$, since the minimum value of γ is 1. In this case, the loss reduces to the original contrastive objective in Equation (4.3).

A few interesting observations from the plot are as follows: First, when $\gamma = 1$, the model is trained using the contrastive loss only, and as we showed is an ineffective objective for generative model learning. This is verified again in Figure C.3 — when $\gamma = 1$, both coherences and latent accuracies take on extremely low values; interestingly, there is a significant boost of performance across all metrics by simply increasing the value of γ from 1 to 1.1; after that, as the value of γ increases, performance on most metrics decreases monotonically (joint coherence being the only exception), and eventually converges to baseline MMVAE (dotted lines in

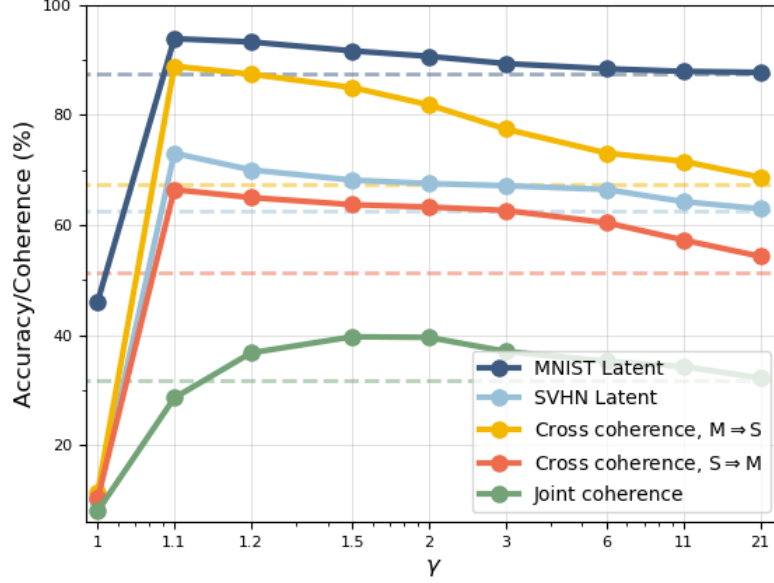


Figure C.3: Performance on different metrics for different values of γ . Dotted lines represents the performance of baseline MMVAE.

Figure C.3). This is unsurprising, since the final objective in Equation (4.4) reduces to the original joint ELBO as γ approaches infinity.

Figure C.3 seems to suggest that 1.1 is the optimal value for hyperparameter γ , however close inspection of the qualitative generative results shows that this might not be the case. See Figure C.4 for a comparison of the model’s generation between MMVAE models trained on (from left to right) $\gamma = 1.1$, $\gamma = 2$ and $\gamma = +\infty$ (i.e. original MMVAE). Although $\gamma = 1.1$ yields model with high coherence scores, we can clearly see from the left-most column of Figure C.4 that the generation of the model seems deprecated, especially for the SVHN modality, where the backgrounds of model’s generation appear to be unnaturally spotty and deformed. This problem is mitigated by increasing γ — as shown in Figure C.3, the image generation quality of $\gamma = 2$ (middle column) is not visibly different from that of $\gamma = +\infty$ (right column).

To verify this observation, we also compute the marginal log likelihood $\log p_{\Theta}(\mathbf{x}, \mathbf{y})$ to quantify the quality of generations. We compute this for all γ s considered in

Figure C.3, and take the average over the entire test set. From the results in Figure C.5, we can see a significant increase of the log likelihood between $\gamma = 1.1$ to $\gamma = 1$. This gain in image generation quality then slows down as γ further increases, and as all other metrics converges to the original MMVAE model.

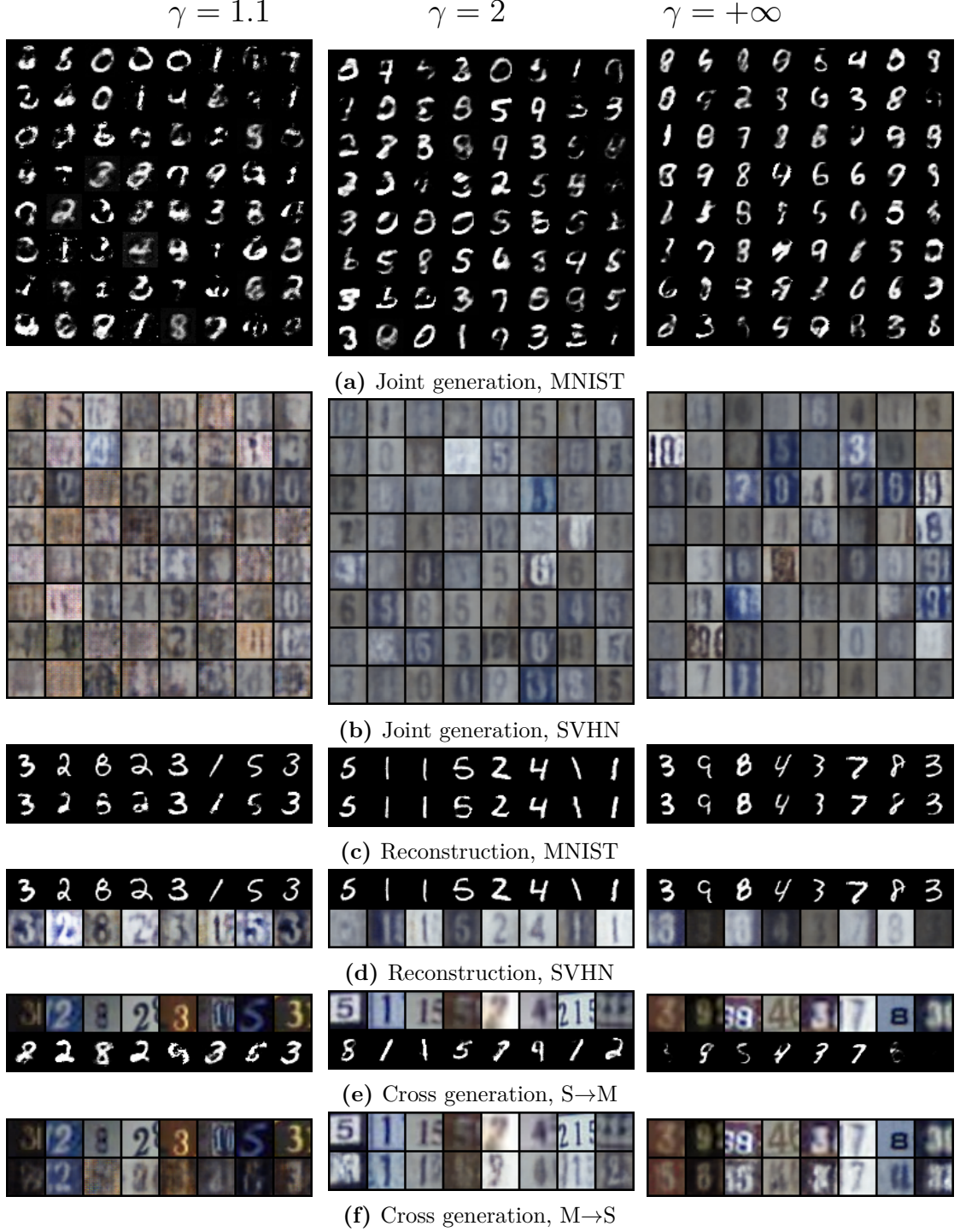


Figure C.4: Generations of MMVAE model trained using the final contrastive objective, with (from left to right) $\gamma = 1.1$, 2 and $+\infty$. Note in (c), (d), (e), (f), the top rows are the inputs and the bottom rows are their corresponding reconstruction/cross generation.

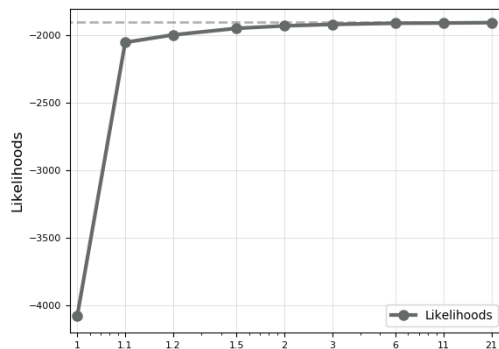


Figure C.5: Performance on different metrics for different values of γ . Dotted lines represents the performance of baseline MMVAE.

C.6 Architecture

We use architectures listed in Table C.1 for the unimodal encoder and decoder for MMVAE, MVAE and JMVAE. For JMVAE we use an extra joint encoder, the architecture of which is described in Table C.2.

Encoder	Decoder
Input $\in \mathbb{R}^{1 \times 28 \times 28}$	Input $\in \mathbb{R}^L$
FC. 400 ReLU	FC. 400 ReLU
FC. L , FC. L	FC. $1 \times 28 \times 28$ Sigmoid

(a) MNIST dataset

Encoder
Input $\in \mathbb{R}^{3 \times 32 \times 32}$
4x4 conv. 32 stride 2 pad 1 & ReLU
4x4 conv. 64 stride 2 pad 1 & ReLU
4x4 conv. 128 stride 2 pad 1 & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1 pad 0
Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. 128 stride 1 pad 0 & ReLU
4x4 upconv. 64 stride 2 pad 1 & ReLU
4x4 upconv. 32 stride 2 pad 1 & ReLU
4x4 upconv. 3 stride 2 pad 1 & Sigmoid

(b) SVHN dataset.

Encoder	Decoder
Input $\in \mathbb{R}^{2048}$	Input $\in \mathbb{R}^L$
FC. 1024 ELU	FC. 256 ELU
FC. 512 ELU	FC. 512 ELU
FC. 256 ELU	FC. 1024 ELU
FC. L , FC. L	FC. 2048

(c) CUB image dataset.

Encoder
Input $\in \mathbb{R}^{1590}$
Word Emb. 128
4x4 conv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 128 stride 2 pad 1 & BatchNorm2d & ReLU
1x4 conv. 256 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
1x4 conv. 512 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1 pad 0
Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. 512 stride 1 pad 0 & ReLU
1x4 upconv. 256 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
1x4 upconv. 128 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
4x4 upconv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 upconv. 1 stride 2 pad 1 & ReLU
Word Emb. ^T 1590

(d) CUB-Language dataset.

Table C.1: Unimodal encoder and decoder architectures.

Encoder
Input $\in \mathbb{R}^{3 \times 32 \times 64}$
4x4 conv. 32 stride 2 pad 1 & ReLU
4x4 conv. 64 stride 2 pad 1 & ReLU
4x4 conv. 128 stride 2 pad 1 & ReLU
1x4 conv. 128 stride 1x2 pad 0x1 & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1 pad 0
(a) MNIST-SVHN dataset.
Encoder
Input $\in \mathbb{R}^{1 \times 32 \times 192}$
4x4 conv. 32 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 64 stride 2 pad 1 & BatchNorm2d & ReLU
4x4 conv. 128 stride 2 pad 1 & BatchNorm2d & ReLU
1x4 conv. 256 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
1x4 conv. 512 stride 1x2 pad 0x1 & BatchNorm2d & ReLU
4x4 conv. L stride 1 pad 0, 4x6 conv. L stride 1 pad 0
(b) CUB Image-Caption dataset.

Table C.2: Joint encoder architectures.

C.7 Qualitative Results on MNIST-SVHN

C.7.1 Generative Results & Marginal Likelihoods on MNIST-SVHN

MMVAE

		$\log p(\mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m)$	$\log p(\mathbf{x}_m \mathbf{x}_n)$
$m = \text{MNIST},$ $n = \text{SVHN}$	MMVAE	-1879.00	-388.59	-388.59	-1618.53
	cI-MMVAE	-1904.15	-385.18	-385.18	-1620.77
	cC-MMVAE	-1924.20	-391.88	-391.84	-1619.34
$m = \text{SVHN},$ $n = \text{MNIST}$	MMVAE	-1879.00	-1472.44	-1472.45	-431.66
	cI-MMVAE	-1904.15	-1491.55	-1491.56	-444.28
	cC-MMVAE	-1924.20	-1490.56	-1494.75	-428.29

Table C.3: Evaluating log likelihoods using original model (MMVAE), contrastive loss with IWAE estimator (cI-MMVAE) and contrastive loss with CUBO estimator (cC-MMVAE). Likelihoods are estimated with IWAE estimator using 1000 samples.

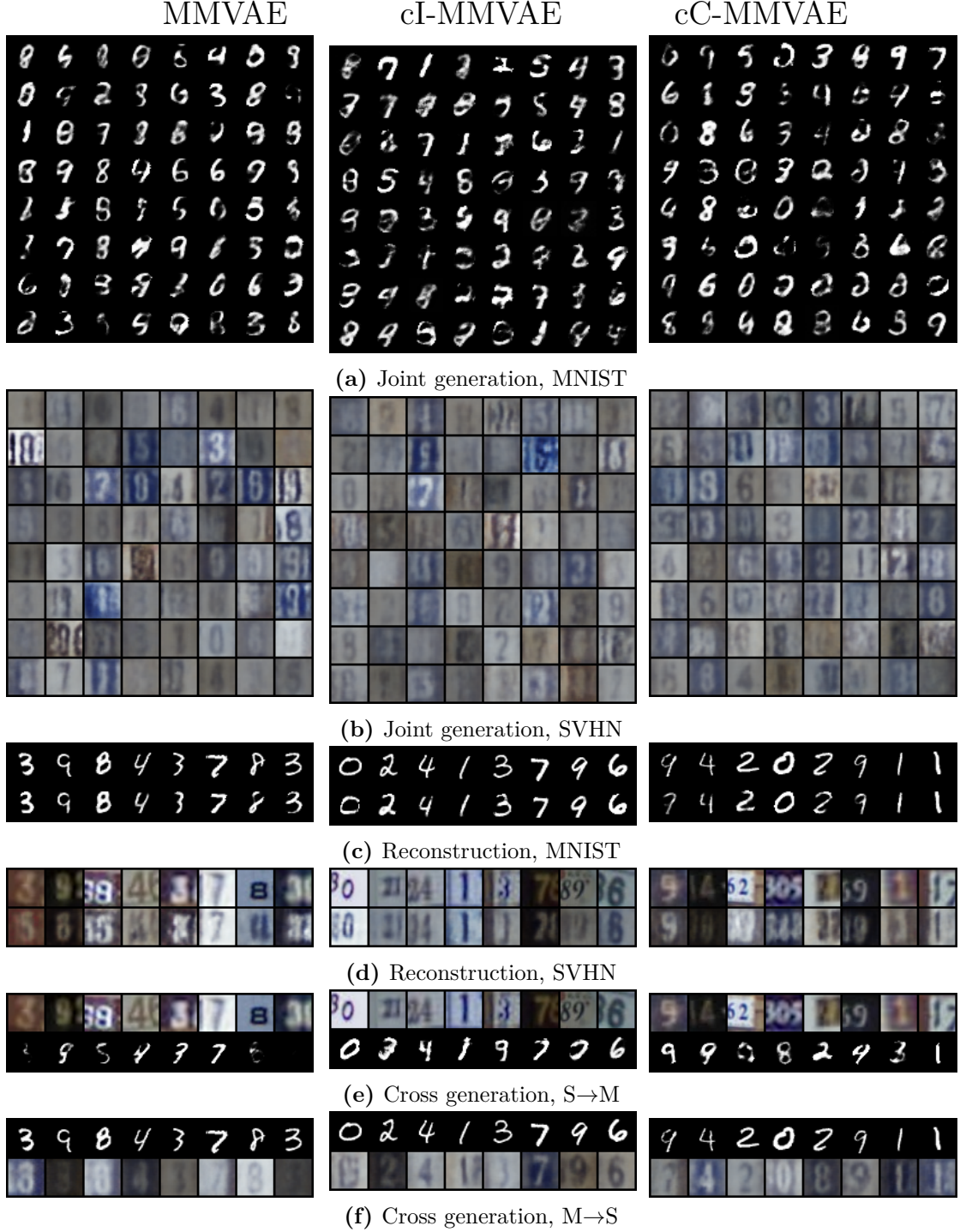


Figure C.6: Generations of MMVAE model, from left to right are original model (MMVAE), contrastive loss with IWAE estimator (cI-MMVAE) and contrastive loss with CUBO estimator (cC-MMVAE).

MVAE

		$\log p(\mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m)$	$\log p(\mathbf{x}_m \mathbf{x}_n)$
$m = \text{MNIST},$ $n = \text{SVHN}$	MVAE	-404.43	-404.43	-388.27	-1847.85
	cI-MVAE	-406.85	-406.22	-388.26	-1876.95
	cC-MVAE	-405.24	-432.96	-388.26	-1889.35
$m = \text{SVHN},$ $n = \text{MNIST}$	MVAE	-404.43	-1518.00	-1488.21	-440.32
	cI-MVAE	-406.85	-1529.32	-1498.73	-443.04
	cC-MVAE	-405.24	-1520.23	-1499.47	-441.01

Table C.4: Evaluating log likelihoods using original model (MVAE), contrastive loss with IWAE estimator (cI-MVAE) and contrastive loss with CUBO estimator (cC-MVAE). Likelihoods are estimated with IWAE estimator using 1000 samples.

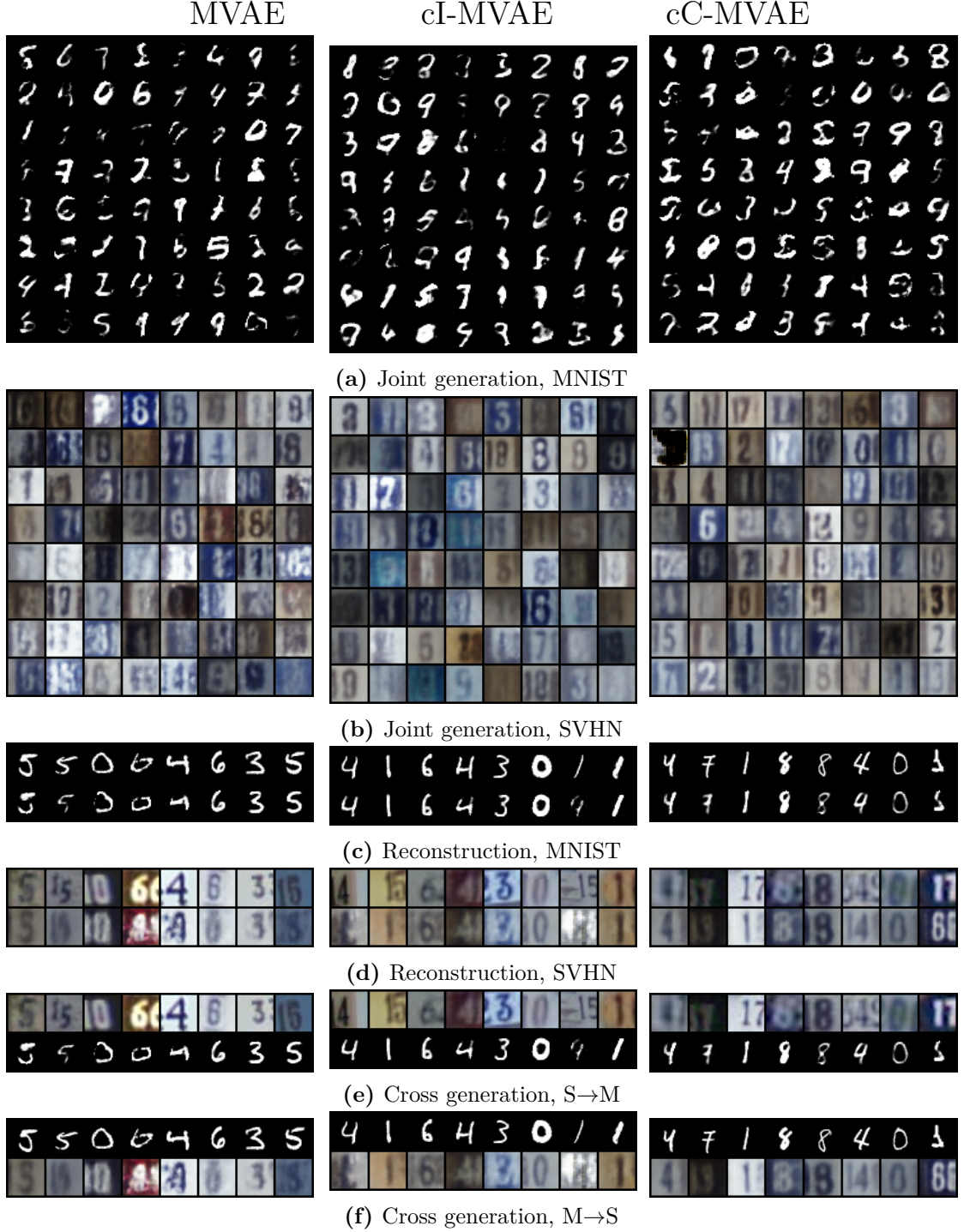


Figure C.7: Generations of MVAE model, from left to right are original model (MVAE), contrastive loss with IWAE estimator (cI-MVAE), contrastive loss with CUBO estimator (cC-MVAE).

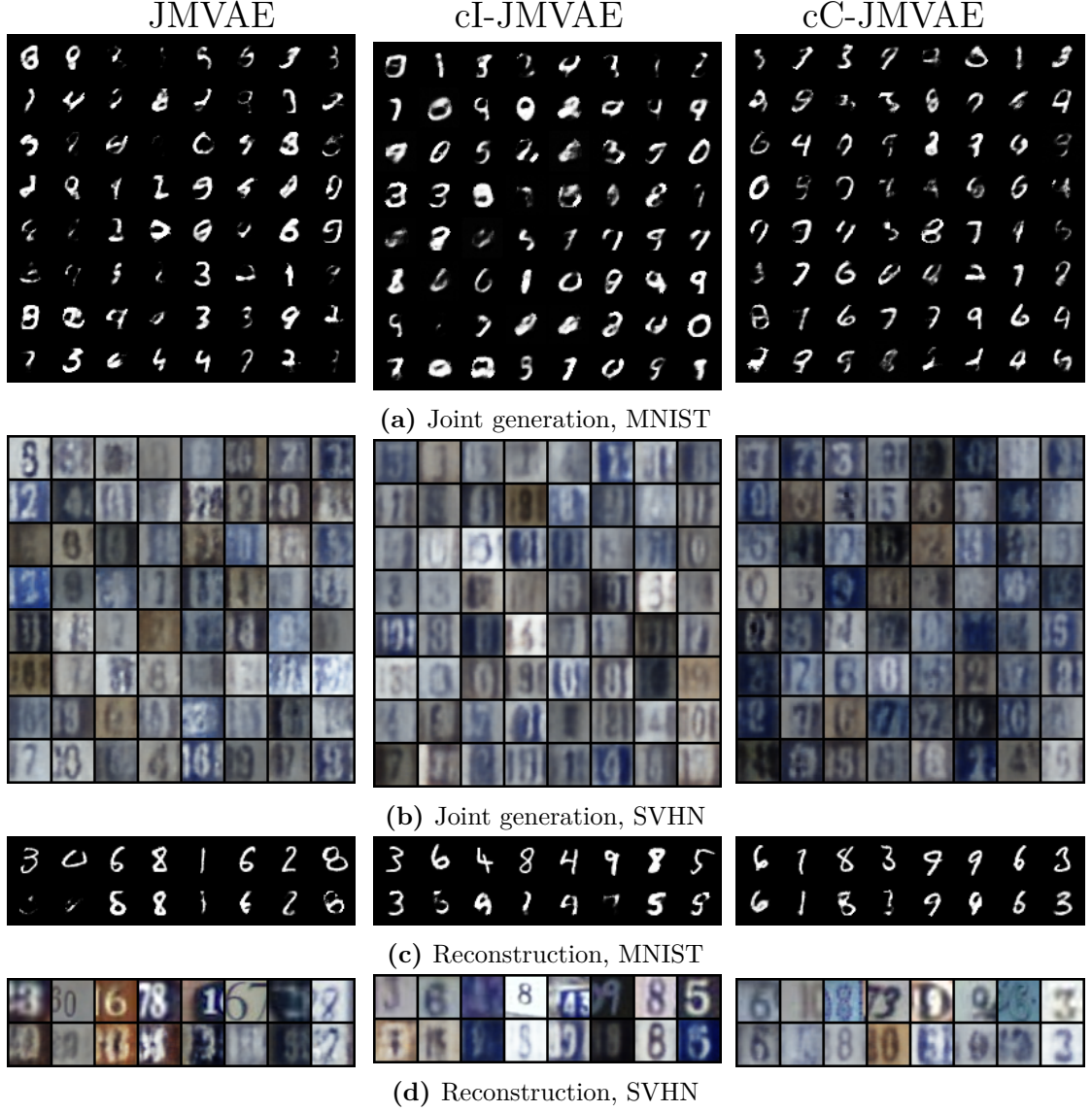


Figure C.8: Generations of JMVAE model, from left to right are original model (JMVAE), contrastive loss with IWAE estimator (cI-JMVAE), contrastive loss with CUBO estimator (cC-JMVAE).

JMVAE

		$\log p(\mathbf{x}_m, \mathbf{x}_n)$	$\log p(\mathbf{x}_m \mathbf{x}_m, \mathbf{x}_n)$
$m = \text{MNIST},$ $n = \text{SVHN}$	JMVAE	-515.44	-497.95
	cI-JMVAE	-518.56	-511.28
	cC-JMVAE	-534.26	-510.90
$m = \text{SVHN},$ $n = \text{MNIST}$	JMVAE	-515.44	-1515.73
	cI-JMVAE	-518.56	-1529.44
	cC-JMVAE	-534.26	-1614.78

Table C.5: Evaluating log likelihoods using original model (JMVAE), contrastive loss with IWAE estimator (cI-JMVAE) and contrastive loss with CUBO estimator (cC-JMVAE). Likelihoods are estimated with IWAE estimator using 1000 samples.

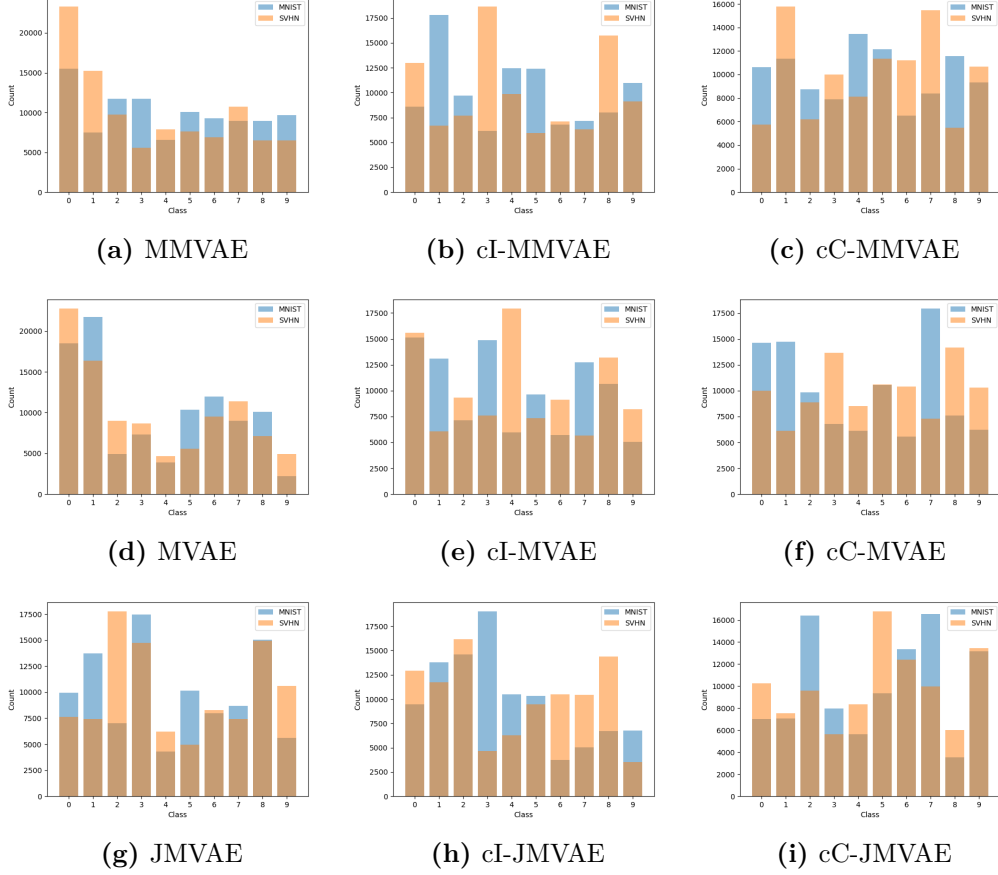


Figure C.9: Number of examples generated for each of class during joint generation.

C.7.2 Generation diversity

To further demonstrate that the improvements from our contrastive objective did not come at a price of sacrificing generation diversity, in Figure C.9 we show histograms of the number of examples generated for each class with samples from prior. Similar to how we compute joint coherence, the class label of generated images are determined using classifiers trained on the original MNIST and SVHN dataset.

We can see that the contrastive models (cI-* and cI-*) are capable of generating examples from different classes, and for MMVAE and MVAE the histogram of contrastive models are more uniform than the original models (less variance between class counts).

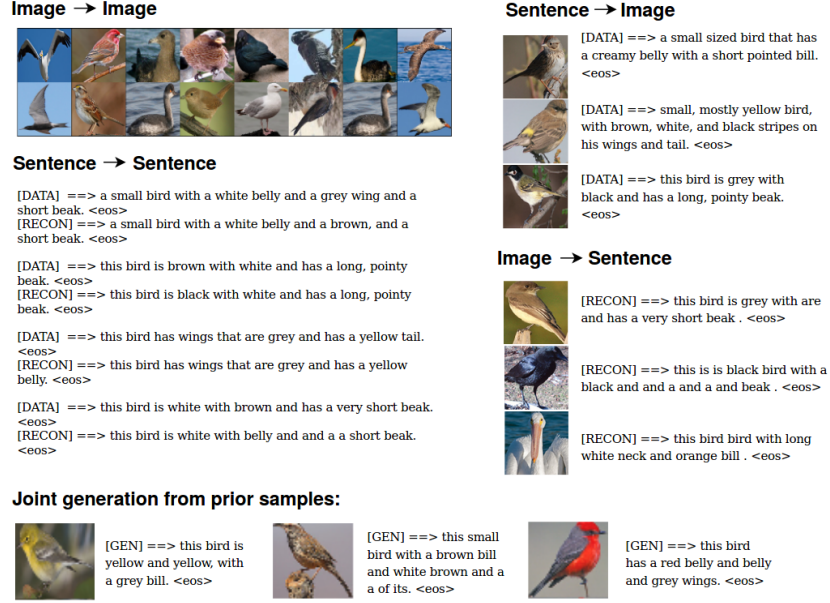


Figure C.10: Qualitative results of MMVAE on CUB Image-Caption dataset, including reconstruction (vision → vision, language → language), cross generation (vision → language, language → vision) and joint generation from prior samples.

C.8 Qualitative results on CUB

The generative results of MMVAE, cI-MMVAE and cC-MMVAE on CUB Image-Caption dataset are as shown in Figure C.10, fig. C.11 and fig. C.12. Note that for the generation in the vision modality, we reconstruct and generate features from ResNet101 and perform nearest neighbour search in all the features in train set to showcase our generation results.

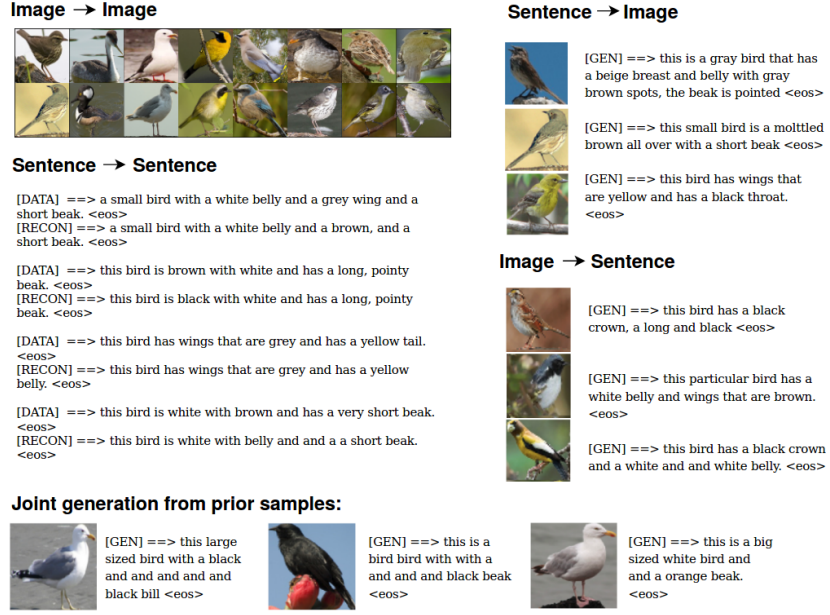


Figure C.11: Qualitative results of MMVAE trained with contrastive loss with IWAE estimator on CUB Image-Caption dataset, including reconstruction (vision → vision, language → language), cross generation (vision → language, language → vision) and joint generation from prior samples.

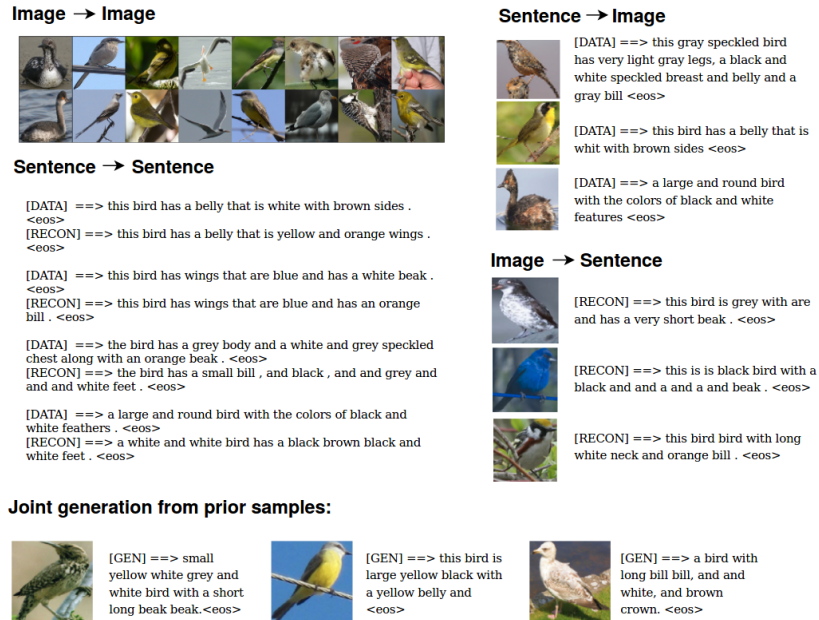


Figure C.12: Qualitative results of MMVAE trained with contrastive loss with CUBO estimator on CUB Image-Caption dataset, including reconstruction (vision → vision, language → language), cross generation (vision → language, language → vision) and joint generation from prior samples.

D

Appendix: Gradient Matching for Domain Generalization

D.1 Taylor Expansion of Reptile and Fish Inner-loop Update

In this section we provide proof to Theorem 5.3.1. We reproduce and adapt the proof from Nichol et al. (2018) in the context of Fish, for completeness.

We demonstrate that when the inner-loop learning rate α is small, the direction of G_f aligns with that of G_g , where

$$G_f = \mathbb{E}[(\theta - \tilde{\theta})] - \alpha S \cdot \bar{G}, \quad (\text{D.1})$$

$$G_g = -\partial \hat{G} / \partial \theta, \quad (\text{D.2})$$

Expanding G_g G_g is the gradient of maximizing the gradient inner product (GIP).

$$G_g = -\frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial}{\partial \theta} G_i \cdot G_j \quad (\text{D.3})$$

Expanding G_f To write out G_f , we need to derive the gradient update of Fish, $\theta - \tilde{\theta}$. Let us first define some notations.

For each inner-loop with S steps of gradient updates, we assume a loss functions l as well as a sequence of inputs $\{d_i\}_{i=1}^S$, where $d_i := \{x_b, y_b\}_{b=1}^B$ denotes a minibatch at step i randomly drawn from one of the available domains in $\{\mathcal{D}_1, \dots, \mathcal{D}_S\}$. For reasons that will become clearer later, take extra note that the subscript i here denotes the index of step, rather than the index of domain. We also define the following:

$$\tilde{g}_i = \mathbb{E}_{d_i} \left[\frac{\partial l((x, y); \theta_i)}{\partial \theta_i} \right] \quad (\text{gradient at step } i, \text{ wrt } \theta_i) \quad (\text{D.4})$$

$$\theta_{i+1} = \theta_i - \alpha \tilde{g}_i \quad (\text{sequence of parameters}) \quad (\text{D.5})$$

$$g_i = \mathbb{E}_{d_i} \left[\frac{\partial l((x, y); \theta_1)}{\partial \theta_1} \right] \quad (\text{gradient at step } i, \text{ wrt } \theta_1) \quad (\text{D.6})$$

$$H_i = \mathbb{E}_{d_i} \left[\frac{\partial^2 l((x, y); \theta_1)}{\partial \theta_1^2} \right] \quad (\text{Hessian at initial point}) \quad (\text{D.7})$$

In the following analysis we omit the expectation $\mathbb{E}_{d_i}$ and input (x, y) to l and instead denote the loss at step i as l_i . Performing second-order Taylor approximation to $\tilde{g}_i$ yields:

$$\tilde{g}_i = l'_i(\theta_i) \quad (\text{D.8})$$

$$= l'_i(\theta_1) + l''_i(\theta_1)(\theta_i - \theta_1) + \underbrace{\mathcal{O}(\|\theta_i - \theta_1\|^2)}_{=\mathcal{O}(\alpha^2)} \quad (\text{D.9})$$

$$= g_i + H_i(\theta_i - \theta_1) + \mathcal{O}(\alpha^2) \quad (\text{D.10})$$

$$= g_i - \alpha H_i \sum_{j=1}^{i-1} \tilde{g}_j + \mathcal{O}(\alpha^2). \quad (\text{D.11})$$

Applying first-order Taylor approximation to $\tilde{g}_j$ gives us

$$\tilde{g}_j = g_j + \mathcal{O}(\alpha), \quad (\text{D.12})$$

plugging this back to Equation (D.11) yields:

$$\tilde{g}_i = g_i - \alpha H_i \sum_{j=1}^{i-1} g_j + \mathcal{O}(\alpha^2). \quad (\text{D.13})$$

For simplicity reason, let us consider performing two steps in inner-loop updates, i.e. $S = 2$. We can then write the gradient of Fish $\theta - \tilde{\theta}$ as

$$\theta - \tilde{\theta} = \alpha(\tilde{g}_1 + \tilde{g}_2) \quad (\text{D.14})$$

$$= \alpha \underbrace{(g_1 + g_2)}_{\textcircled{1}} - \alpha^2 \underbrace{H_2 g_1}_{\textcircled{2}} + \mathcal{O}(\alpha^3). \quad (\text{D.15})$$

Furthermore, taking the expectation of $\theta - \tilde{\theta}$ under minibatch sampling gives us (assuming independence between g_1 and g_2)

$$\begin{aligned}
\textcircled{1} &= \mathbb{E}_{1,2} [g_1 + g_2] = G_1 + G_2 \\
\textcircled{2} &= \mathbb{E}_{1,2} [H_2 g_1] = \mathbb{E}_{1,2} [H_1 g_2] && \text{(interchanging indices)} \\
&= \frac{1}{2} \mathbb{E}_{1,2} [H_2 g_1 + H_1 g_2] && \text{(averaging last two eqs)} \\
&= \frac{1}{2} \mathbb{E}_{1,2} \left[\frac{\partial(g_1 \cdot g_2)}{\partial \theta_1} \right] \\
&= \frac{1}{2} \cdot \frac{\partial(G_1 \cdot G_2)}{\partial \theta_1}.
\end{aligned}$$

Note that the only reason we can interchange the indices in $\textcircled{2}$ is because the subscripts represent steps in the inner loop rather than index of domains. Plugging $\textcircled{1}$, $\textcircled{2}$ in Equation (D.15) yields:

$$\mathbb{E}[\theta - \tilde{\theta}] = \alpha(G_1 + G_2) - \frac{\alpha^2}{2} \cdot \frac{\partial(G_1 \cdot G_2)}{\partial \theta_1} + \mathcal{O}(\alpha^3) \quad (\text{D.16})$$

We can also expand this to the general case where $S \geq 2$:

$$\begin{aligned}
\mathbb{E}[\theta - \tilde{\theta}] &= \alpha \sum_{s=1}^S G_s - \frac{\alpha^2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial(G_i \cdot G_j)}{\partial \theta_1} + \mathcal{O}(\alpha^3).
\end{aligned} \quad (\text{D.17})$$

The second term in Equation (D.1) is $\bar{G}$, which is the full gradient of ERM defined as follow:

$$\bar{G} = \frac{1}{S} \sum_{s=1}^S G_s. \quad (\text{D.18})$$

Plugging Equation (D.17) and Equation (D.18) to Equation (D.1) yields

$$G_f = \mathbb{E}[\theta - \tilde{\theta}] - \alpha S \bar{G} \quad (\text{D.19})$$

$$= -\frac{\alpha^2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial}{\partial \theta_1} G_i \cdot G_j \quad (\text{D.20})$$

Comparing Equation (D.3) to Equation (D.20), we have:

$$\lim_{\alpha \rightarrow 0} \frac{G_f \cdot G_g}{\|G_f\| \|G_g\|} = 1.$$

D.1.1 Fish and Reptile: Differences and Connections

As we introduced, our algorithm Fish is inspired by Reptile, a model agnostic meta-learning (MAML) algorithm.

Meta-learning aims at reducing the sample complexity of new, unseen tasks. A popular school of thinking in meta-learning is MAML, first proposed in Finn et al. (2017); Andrychowicz et al. (2016). The key idea is to backpropagate through gradient descent itself to learn representations that can be easily adapted to unseen tasks. There are close connections between meta-learning (Thrun and Pratt, 1998) and (multi-source) domain adaptation. In fact, there are a few works in domain generalization that are inspired by the meta-learning principles, such as Li et al. (2018a); Balaji et al. (2018); Li et al. (2019); Dou et al. (2019). In Ren et al. (2018), we also see the leveraging of gradient inner product in meta-learning, where it is used to determine the importance weight of training examples.

Even though meta learning and domain generalization both study N -way, K -shot problems, there are some distinct differences that set them apart. The most prominent one is that in meta learning, some examples in the test dataset will be made available at test time ($K > 0$), while in domain generalization no example in the test dataset is seen by the model ($K = 0$); another important difference is that while domain generalization aims to train models that perform well on an unseen distribution of the *same task*, meta-learning assumes *multiple tasks* and requires the model to quickly learn an unseen task using only K examples.

Due to these differences, it does not make sense in general to use MAML framework in domain generalization. As it turns out however, the idea of aligning gradients to improve generalization is relevant to both methods — The fundamental difference here that MAML algorithms such as Reptile aligns the gradients between batches from the *same* task Nichol et al. (2018), while Fish aligns those between batches from *different* tasks.

Algorithm 4: Black fonts denote steps used in **both algorithms**, colored fonts are steps unique to **Fish** or **Reptile**.

```

1: for  $i = 1, 2, \dots$  do
2:    $\tilde{\theta} \leftarrow \theta$ 
3:   Sample task  $\mathcal{D}_t \sim \{\mathcal{D}_1, \dots, \mathcal{D}_T\}$ 
4:   for  $s \in \{1, \dots, S\}$  or  $\mathcal{D}_t \in \{\mathcal{D}_1, \dots, \mathcal{D}_T\}$  do
5:     Sample batch  $\mathbf{d}_t \sim \mathcal{D}_t$ 
6:      $g_t = \partial \mathcal{L}(\mathbf{d}_t; \tilde{\theta}) / \partial \tilde{\theta}$ 
7:     Update  $\tilde{\theta} \leftarrow \tilde{\theta} - \alpha g_t$ 
8:   end for
9:   Update  $\theta \leftarrow \theta + \epsilon(\tilde{\theta} - \theta)$ 
10: end for

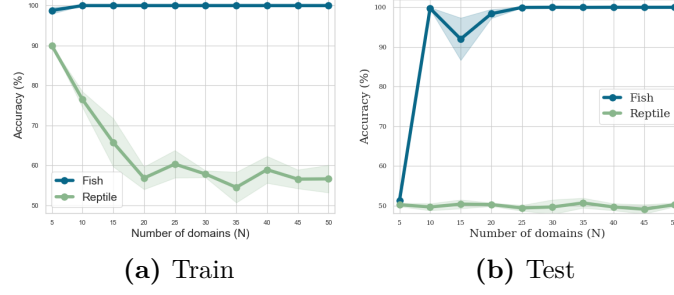
```

To see how this is achieved, let us have a look at the algorithmic comparison between **Fish** (blue) and **Reptile** (green) in Algorithm 4. As we can see, the key difference between the algorithm of Fish and Reptile is that Reptile performs its inner-loop using minibatches from the *same* task, while Fish uses minibatches from *different* tasks (4-8). Based on the analysis in Nichol et al. (2018) (which we reproduce in Appendix D.1), this is why Reptile maximizes the within-task gradient inner products and Fish maximizes the across-task gradient inner products.

A natural question to ask here is – how does this affect their empirical performance? In Figure D.1, we show the train and test performance of **Fish** (blue) and **Reptile** (green) on CDSprites-N. We can see that despite the algorithmic similarity between Fish and Reptile, the two methods behave very differently on this domain generalization task: while Fish’s test accuracy goes to 100% at $N = 10$, Reptile’s test performance is always 50% regardless of N . Moreover, we observe a dip in Reptile’s training performance early on, with the accuracy plateaus at 56% when $N > 20$. Reptile’s poor performance on this dataset is to be expected since its inner-loop is designed to encourage within-domain generalization, which is not helpful for learning what’s invariant across domains.

D.1.2 When $\epsilon = 1$

When we set the meta learning rate of Fish $\epsilon = 1$, Fish reduces to ERM with a data-loader that samples from only one domain in each minibatch. Note that with

**Figure D.1:** Performance on CDSPRITES-N, with $N \in [5, 50]$

this change, the inner-loop step S now represents the total training iterations, and as a result it increases from a constant value (≤ 10 in our experiments) to orders of magnitude larger (up to 10^5 for some datasets that we use). We demonstrate here that for this reason, the effect of maximizing inter-domain gradient inner product is significantly less prominent when $\epsilon = 1$.

To see this, let us revisit Equation (D.17), where we demonstrate that the expectation to Fish’s update $\mathbb{E}[\theta - \tilde{\theta}]$ can be written as the sum of the following three terms via Taylor series expansion,

$$\mathbb{E}[\theta - \tilde{\theta}] = \underbrace{\alpha \sum_{s=1}^S G_s}_{(1)} - \underbrace{\frac{\alpha^2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} \frac{\partial(G_i \cdot G_j)}{\partial \theta_1}}_{(2)} + \underbrace{\mathcal{O}(\alpha^3)}_{(3)}, \quad (\text{D.21})$$

where (1) is the sum of gradients over S steps of SGD updates (equivalent to ERM), (2) the gradient inner product over S steps of gradients and (3) is the higher order terms of Taylor series expansion.

For both Fish and Reptile, the higher order terms are ignored under two conditions: 1) constant/non-infinite inner-loop step S and 2) small learning rate α . However, when $\epsilon = 1$, inner-loop step $S \rightarrow \infty$. As a result, the higher order terms cannot be ignored, and we can no longer conclude that gradient inner product plays an important rule in the model’s updates.

D.2 SmoothFish: a more general algorithm

D.2.1 Derivation

We conclude in Appendix D.1 that a component of Fish’s update $G_f = \mathbb{E}[\theta - \tilde{\theta}] - \alpha S \cdot \bar{G}$ is in the same direction as the gradient of GIP, G_g . It is therefore possible to have explicit control over the scaling of the GIP component in Fish, similar to the original IDGM objective, by writing the following:

$$G_{\text{sm}} = \alpha S \cdot \bar{G} + \gamma \left(\mathbb{E}[\theta - \tilde{\theta}] - \alpha S \cdot \bar{G} \right). \quad (\text{D.22})$$

By introducing the scaling term γ , we have better control on how much the objective focus on inner product vs average gradient. Note that $\gamma = 1$ recovers the original Fish gradient, and when $\gamma = 0$ the gradient G_{sm} is equivalent to ERM’s gradient with learning rate αS . We name the resulting algorithm SmoothFish. See Algorithm 5.

D.2.2 Results

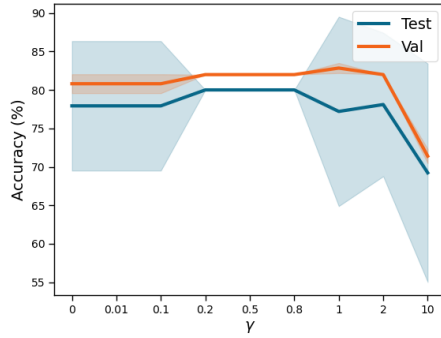
We run experiments on 4 datasets in WILDS using SmoothFish, with γ ranging in $[0, 0.1, 0.2, 0.5, 0.8, 1, 2, 10]$. Note that when $\gamma = 0$, SmoothFish is equivalent to ERM and when $\gamma = 1$ it is equivalent to Fish. See results in Figure D.2. The other hyperparameters including α , meta steps, ϵ used here are the same as the ones used in our main experiments section.

Algorithm 5: Smoothed version of Fish, which allows to get approximate gradients for the general form of Equation (5.4).

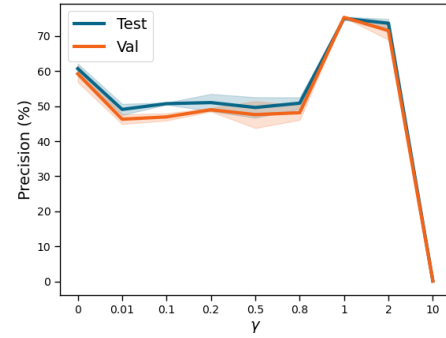
```

1: for iterations = 1, 2, ... do
2:    $\tilde{\theta} \leftarrow \theta$ 
3:   for  $\mathcal{D}_i \in \text{permute}(\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_S\})$  do
4:     Sample batch  $d_i \sim \mathcal{D}_i$ 
5:      $g_i = \mathbb{E}_{d_i} \left[ \frac{\partial l((x, y); \theta)}{\partial \theta} \right]$  // Grad wrt  $\theta$ 
6:      $\tilde{g}_i = \mathbb{E}_{d_i} \left[ \frac{\partial l((x, y); \tilde{\theta})}{\partial \tilde{\theta}} \right]$  // Grad wrt  $\tilde{\theta}$ 
7:     Update  $\tilde{\theta} \leftarrow \tilde{\theta} - \alpha \tilde{g}_i$ 
8:   end for
9:    $\bar{g} = \frac{1}{S} \sum_{s=1}^S g_i$ ,  $g_{\text{sm}} = \alpha S \bar{g} + \gamma ((\tilde{\theta} - \theta) - \alpha S \bar{g})$ 
10:  Update  $\theta \leftarrow \theta + \epsilon g_{\text{sm}}$ 
11: end for

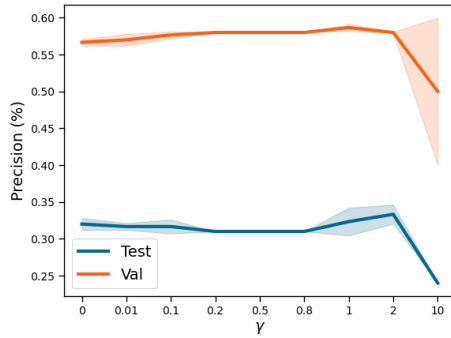
```



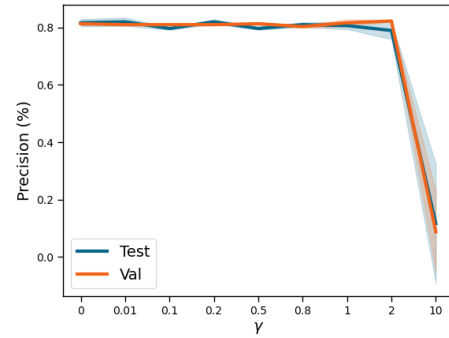
(a) CAMELYON17



(b) CIVILCOMMENTS



(c) FMoW



(d) POVERTY

Figure D.2: Results on WILDS using SmoothFish with γ ranging from 0 to 10.

D.3 CdSprites Architecture

In Table D.1 we outline the architecture of all models trained on CdSprites dataset.

Encoder
Input $\in \mathbb{R}^{3 \times 64 \times 64}$
4x4 conv. 32 stride 2 pad 1 & ReLU
4x4 conv. 32 stride 2 pad 1 & ReLU
4x4 conv. 64 stride 2 pad 1 & ReLU
4x4 conv. 128 stride 2 pad 1 & ReLU
4x4 conv. 128 stride 2 pad 0 & ReLU
fully connected, 3 output channels

Table D.1: Architecture of models trained on CdSprites.

D.4 Discussions and Results on Wilds

We provide a more detailed summary of each dataset in Table D.2.

Some entries in **# Domains** are omitted because the domains in each split overlap. Note that in this paper we report the results on WILDS v1 — the benchmark has been updated since with slightly different dataset splits. We are currently working on updating our results to v2 of WILDS.

Table D.2: Details of the 6 WILDS datasets we experimented on.

Dataset	Domains (# domains)	Data (x)	Target (y)	# Examples			# Domains		
				train	val	test	train	val	test
FMoW	Time (16), Regions (5)	Satellite images	Land use (62 classes)	76,863	19,915	22,108	11, -	3, -	2, -
POVERTY	Countries (23), Urban/rural (2)	Satellite images	Asset (real valued)	10,000	4,000	4,000	13, -	5, -	5, -
CAMELYON17	Hospitals (5)	Tissue slides	Tumor (2 classes)	302,436	34,904	85,054	3	1	1
CIVILCOMMENTS	Demographics (8)	Online comments	Toxicity (2 classes)	269,038	45,180	133,782	-	-	-
iWILDCAM2020	Trap locations (324)	Photos	Animal species (186 classes)	142,202	20,784	38,943	245	32	47
AMAZON	Reviewers (7,676)	Product reviews	Star rating (5 classes)	1,000,124	100,050	100,050	5,008	1,334	1,334

Table D.3: Results on POVERTYMAP-WILDS.

Table D.4: Results on CAMELYON17-WILDS.

Method	Val. Worst-U/R r	Test Worst-U/R r	Method	Val. Accuracy (%)	Test Accuracy (%)
Fish	0.47 (± 0.01)	0.30 (± 0.01)	Fish	83.9 (± 1.2)	74.7 (± 7.1)
IRM	0.53 (± 0.05)	0.43 (± 0.07)	ERM	84.9 (± 3.1)	70.3 (± 6.4)
ERM	0.80 (± 0.04)	0.78 (± 0.04)	ERM (ours)	84.1 (± 2.4)	70.5 (± 12.1)
ERM (ours)	0.48 (± 0.11)	0.29 (± 0.02)	IRM	86.2 (± 1.4)	64.2 (± 8.1)
Coral	0.51 (± 0.06)	0.45 (± 0.06)	Coral	86.2 (± 1.4)	59.5 (± 7.7)

D.4.1 PovertyMap-Wilds

Task: Asset index prediction (real-valued). Domains: 23 countries

The task is to predict the real-valued asset wealth index of an area, given its satellite imagery. Since the number of domains considered here is large (23 countries), instead of looping over all S domains in each inner-loop, we sample $N \ll S$ domains in each iteration and perform inner-loop updates using minibatches from these domains only to speed up computation. For this dataset we choose $N = 5$ by hyper-parameter search.

Evaluation: *Worst-U/R Pearson Correlation (r).* Following the practice in WILDS benchmark, we compare the results by computing the worst region Pearson correlation (r) between the predicted and ground-truth asset index over 3 random seed runs.

Results: We train the model using a ResNet-18 (He et al., 2016) backbone. See Table D.3.

We see that Fish obtains the highest test performance, with the same validation performance as the best baseline. The performance is more stable between validation and test, and the standard deviation is smaller than for the baselines. We also report the results of ERM models trained in our environment as “ERM (ours)”, which shows similar performance to the canonical results reported in the WILDS benchmark itself (“ERM”).

D.4.2 Camelyon17-Wilds

Task: Tumor detection (2 classes). Domains: 5 hospitals

The CAMELYON17-WILDS dataset contains 450,000 lymph-node scans from 5 hospitals. Due to the size of the dataset, instead of training with Fish from scratch, we pre-train the model with ERM using the recommended hyper-parameters in Koh et al. (2020), and fine-tune with Fish. For this dataset, we find that Fish performs the best when starting from a pretrained model that has not yet converged,

achieving much higher accuracy than the ERM model. we provide an ablation study on this in Appendix D.6.

Evaluation: *Average accuracy.* We evaluate the average accuracy of this binary classification task. Following Koh et al. (2020), we show the mean and standard deviation of results over 10 random seeds runs. The number of random seeds required here is greater than other WILDS datasets due to the large variance observed in results. Note that these random seeds are not only applied during the fine-tuning stage, but also to the pretrained models to ensure a fair comparison.

Results: Following the practice in WILDS, we adopt DenseNet-121’s (Huang et al., 2017) architecture for models trained on this dataset. See results in Table D.4.

The results show that Fish significantly outperforms all baselines – its test accuracy surpasses the best performing baseline by 6%. Also note that for all other baselines, there is a large gap between validation and test accuracy (11%~27%). This is because WILDS chose the hospital that is the most difficult to generalize to as the test split to make the task more challenging. Surprisingly, as we can observe in Table D.4, the discrepancy between test and validation accuracy of Fish is quite small (3%). The fact that it is able to achieve a similar level of accuracy on the worst-performing domain further demonstrates that Fish does not rely on domain-specific information, and instead makes predictions using the invariant features across domains.

To demonstrate that Fish’s strong performance on CAMELYON17’s selected test set is not merely coincidental, we randomly chose 3 ways of assigning the 5 domains as train/test/val (3/1/1 domains) splits. See results collected over 10 random seeds in Table D.5.

Notably, on these different splits of CAMELYON17 Fish still on average outperforms ERM (with only 2 exceptions), demonstrating that Fish can generally achieve good performance on the Camelyon17 dataset.

Table D.5: Results on CAMELYON17-WILDS with shuffled splits.

Train/Val/Test ID	Val, ERM (%)	Val, Fish (%)	Test, ERM (%)	Test, Fish (%)
012/3/4	71.3 (± 6.3)	71.9 (± 5.1)	80.9 (± 8.6)	88.9 (± 6.9)
124/0/3	72.3 (± 5.3)	74.1 (± 4.3)	63.4 (± 6.6)	65.9 (± 3.9)
234/0/1	83.9 (± 3.8)	84.1 (± 2.3)	75.6 (± 2.8)	73.9 (± 3.5)
034/1/2 (original)	84.3 (± 2.1)	82.5 (± 1.2)	73.3 (± 9.0)	79.5 (± 6.0)

Table D.6: Results on FMoW-WILDS.

Method	Val. Accuracy (%)		Test Accuracy (%)	
	Average	Worst	Average	Worst
Fish	57.8 (± 0.15)	49.5 (± 2.34)	51.8 (± 0.32)	34.6 (± 0.18)
ERM	59.5 (± 0.37)	48.9 (± 0.62)	53.0 (± 0.55)	32.3 (± 1.25)
ERM (ours)	59.9 (± 0.22)	47.1 (± 1.21)	52.9 (± 0.18)	30.9 (± 1.53)
IRM	57.4 (± 0.37)	47.5 (± 1.57)	50.8 (± 0.13)	30.0 (± 1.37)
Coral	56.9 (± 0.25)	47.1 (± 0.43)	50.5 (± 0.36)	31.7 (± 1.24)

Table D.7: Results on CIVILCOMMENTS-WILDS.

Method	Val. Accuracy (%)		Test Accuracy (%)	
	Average	Worst	Average	Worst
Fish	88.8 (± 0.6)	70.5 (± 1.0)	89.4 (± 0.2)	75.3 (± 0.6)
Group DRO	89.6 (± 0.3)	68.7 (± 1.0)	89.4 (± 0.3)	70.4 (± 2.1)
Rewighted	89.1 (± 0.3)	67.9 (± 1.2)	88.9 (± 0.3)	67.3 (± 0.1)
ERM	92.3 (± 0.6)	53.6 (± 0.7)	92.2 (± 0.6)	58.0 (± 1.2)
ERM (ours)	92.1 (± 0.5)	54.1 (± 0.4)	92.5 (± 0.3)	58.1 (± 1.7)

D.4.3 FMoW-Wilds

Task: Infrastructure classification (62 classes). Domains: 80 (16 years $\times$ 5 regions)

Similar to CAMELYON17-WILDS, since the number of domains is large, we sample $N = 5$ domains for each inner-loop. To speed up computation, we also use a pretrained ERM model and fine-tune with Fish; different from Appendix D.4.2, we find the best-performing models are acquired when using converged pretrained models (see details in Appendix D.6).

Evaluation: *Average & worst-region accuracies.* Following WILDS, the average accuracy evaluates the model’s ability to generalize over years, and the worst-region accuracy measures the model’s performance across regions under a time shift. We report results using 3 random seeds.

Results: Following Koh et al. (2020), we use a DenseNet-121 pretrained on ImageNet for this dataset. Results in Table D.6 show that Fish has the highest worst-region accuracy on both test and validation sets. It ranks second in terms of average accuracy, right after ERM. Again, Fish’s performance is notably stable with the smallest standard deviation across all metrics compared to baselines.

D.4.4 CivilComments-Wilds

Task: Toxicity detection in online comments (2 classes). Domains: 8 demographic identities.

The CIVILCOMMENTS-WILDS contains 450,000 comments collected from online articles, each annotated for toxicity and the mentioning of demographic identities. Again, we use ERM pre-trained model to speed up computation, and sample $N = 5$ domains for each inner-loop.

Evaluation: *Worst-group accuracy.* To study the bias of annotating comments that mentions demographic groups as toxic, the WILDS benchmark proposes to evaluate the model’s performance by doing the following: 1) Further separate each of the 8 demographic identities into 2 groups by toxicity – for example, separate *black* into *black, toxic* and *black, not toxic*; 2) measure the accuracies of these $8 \times 2 = 16$ groups and use the lowest accuracy as the final evaluation of the model. This metric is equivalent to computing the sensitivity and specificity of the classifier on each demographic identity, and reporting the worse of the two metrics over all domains. Good performance on the group with the worst accuracy implies that the model does not tend to use demographic identity as an indicator of toxic comments.

Again, following Koh et al. (2020) we report results of 3 random seed runs.

Results: We compare results to the baselines used in the WILDS benchmark over 3 random seed runs in Table D.7. All models are trained using DistilBERT (Sanh et al., 2019).

The results show that Fish outperforms the best baseline by 4% and 7% on the test and validation set’s worst-group accuracy respectively, and is competitive in terms of average accuracy with ERM (within standard deviation). The strong performance of Fish on worst-group accuracy suggests that the model relies the least on demographic identity as an indicator of toxic comments compared to other baselines. ERM, on the other hand, has the highest average accuracy and the lowest worst-group accuracy. This indicates that it achieves good average performance

by leveraging the spurious correlation between toxic comments and the mention of certain demographic groups.

Note that different from all other datasets in WILDS that focus on *pure domain generalization* (i.e., no overlap between domains in train and test splits), CIVILCOMMENTS-WILDS is a *subpopulation shift* problem, where the domains in test are a subpopulation of the domains in train. As a result, the baseline models used in WILDS for this dataset are different from the methods used in all other datasets, and are tailored to avoiding systematic failure on data from minority subpopulations. Fish works well in this setting too without any changes or special sampling strategies (such as * and + in Table D.7). This further demonstrates the good performance of our algorithm on different domain generalization scenarios.

D.4.5 iWildCam-Wilds

Task: Animal species (186 classes). Domains: 324 camera locations.

The dataset consists of over 200,000 photos of animal in the wild, using stationary cameras across 324 locations. Classifying animal species from these heat or motion-activated photos is especially challenging: methods can easily rely on the background information of photos from the same camera setup. Fish models are pretrained with ERM till convergence, and for each inner loop we sample from $N = 10$ domains.

Evaluation: *Macro F1 score.* Across the 186 class labels, we report average accuracy and both weighted and macro F1 scores (F1-w and F1-m, respectively, in Table D.8). We run 3 random seeds for each model.

Results: All models reported in Table D.4 are trained using a ResNet-50. We find Fish to outperform baselines on both test accuracy and weighted F1, with a 1% improvement on both metrics over the best performing model (ERM). However, this comes at the cost of lower macro F1 score, where Fish performs 1% worse than ERM models that we trained and 3% than the ERM reported in WILDS.

This suggests that Fish is less good at classifying rarer species, however the overall accuracy on the test dataset is improved.

Although Fish did not outperform the ERM baseline on the primary evaluation metric proposed in Koh et al. (2020), we found the improvement of Fish in both accuracy and weighted F1 to be robust across a range of hyperparameters. See more details on this in Appendix D.5.

D.4.6 Amazon-Wilds

Task: Sentiment analysis (5 classes). Domains: 7,676 Amazon reviewers.

The dataset contains 1.4 million customer reviews on Amazon from 7,676 customers, and the task is to predict the score (1-5 stars) given the review. Similarly, we pretrained the model with ERM till convergence, and due to the large number of domains ($S = 5008$ in train) we sample $N = 5$ reviewers for each inner loop.

Evaluation: *10th percentile accuracy.* Reporting the accuracy of the 10th percentile reviewer helps us assess whether the model performance is consistent across different reviewers. The results in Table D.9 are reported over 3 random seeds.

Results: The model is trained using DISTILBERT (Sanh et al., 2019) backbone. While Fish has lower average accuracy compared to ERM, its 10th percentile accuracy matches that of ERM, outperforming all other baselines.

Table D.8: Results on IWILDCAM-WILDS.

Method	Test ID Macro F1 (%)	Test ID Avg Acc (%)	Test OOD Macro F1 (%)	Test OOD Avg Acc (%)
Fish	40.3 (± 0.6)	73.8 (± 0.1)	22.0 (± 1.8)	64.7 (± 2.6)
GroupDRO	37.5 (± 1.7)	71.6 (± 2.7)	23.9 (± 2.1)	72.7 (± 2.0)
ERM	47.0 (± 1.4)	75.7 (± 0.3)	31.0 (± 1.3)	71.6 (± 2.5)
Coral	43.5 (± 3.5)	73.7 (± 0.4)	32.8 (± 0.1)	73.3 (± 4.3)
IRM	22.4 (± 7.7)	59.9 (± 8.1)	15.1 (± 4.9)	59.8 (± 3.7)

Table D.9: Results on AMAZON-WILDS.

Method	Val. Accuracy (%)		Test Accuracy (%)	
	Average	10-th per.	Average	10-th per.
Fish	72.5 (± 0.0)	54.0 (± 0.0)	71.7 (± 0.0)	53.3 (± 0.0)
ERM	72.7 (± 0.1)	55.2 (± 0.7)	71.9 (± 0.1)	53.8 (± 0.0)
IRM	71.5 (± 0.3)	54.2 (± 0.8)	70.5 (± 0.3)	52.4 (± 0.8)
Reweighted	69.1 (± 0.0)	52.1 (± 0.2)	68.6 (± 0.6)	52.0 (± 0.0)

D.5 Hyperparameters

In Table D.10 we list the hyperparameters we used to train ERM. The same hyperparameters were used for producing ERM baseline results and as pretrained models for Fish. In `val. metric` we report the metric on validation set that is used for model selection, and in `cut-off` we specify when to stop training when using ERM to generate pretrained models.

Table D.10: Hyperparameters for ERM. We follow the hyperparameters used in WILDS benchmark. Note that we did not use a pretrained model for POVERTY, therefore its cut-off condition is not reported.

Dataset	Model	Learning rate	Batch size	Weight decay	Optimizer	Val. metric	Cut-off
CAMELYON17	Densenet-121	1e-3	32	0	SGD	acc. avg.	iter 500
CIVILCOMMENTS	DistilBERT	1e-5	16	0.01	Adam	acc. wg.	Best val. metric
FMoW	Densenet-121	1e-4	64	0	Adam	acc. avg.	Best val. metric
iWILDCAM	Resnet-50	1e-4	16	0	Adam	F1-macro (all)	Best val. metric
POVERTY	Resnet-18	1e-3	64	0	Adam	Worst-U/R Pearson (r)	-
AMAZON	DistilBERT	2e-6	8	0.01	Adam	10th percentile acc.	-

In Table D.11 we list out the hyperparameters we used to train Fish. Note that we train Fish using the same model, batch size, val metric and optimizer as ERM – these are not listed in Table D.11 to avoid repetitions. Weight decay is always set as 0.

Table D.11: Hyperparameters for Fish.

Dataset	Group by	α	ϵ	# domains	Meta steps
CAMELYON17	Hospitals	1e-3	0.01	3	3
CIVILCOMMENTS	Demographics $\times$ toxicity	1e-5	0.05	16	5
FMoW	time $\times$ regions	1e-4	0.01	80	5
iWILDCAM	Trap locations	1e-4	0.01	324	10
POVERTY	Countries	1e-3	0.1	23	5
AMAZON	Reviewers	2e-6	0.01	7,676	5

D.6 Ablation Studies on Pre-trained Models

In this section we perform ablation study on the convergence of pretrained ERM models. Note that the pretraining is done on the same domain generalization dataset that Fish is trained on later, not on ImageNet. We study the performance of Fish with the following three configurations of pretrained ERM models:

1. Model is trained on 10% of the data (epoch 1);
2. Model is trained on 50% of the data (epoch 1);
3. Model at convergence.

By comparing the results between these three settings, we demonstrate how the level of convergence affects the Fish’s training performance. See results in Table D.12. Note that POVERTY is excluded here because the dataset is small enough that we are able to train Fish from scratch.

Table D.12: Ablation study on pretrained ERM models.

Model	FMoW	CAMELYON17	iWILDCAM	CIVILCOMMENTS
	Test Avg Acc	Test Avg Acc	Test Macro F1	Test Worst Acc
10% data	21.7 (± 2.5)	79.1 (± 12.3)	13.7 (± 0.5)	71.8 (± 1.3)
50% data	31.0 (± 0.8)	64.6 (± 12.3)	19.0 (± 0.06)	74.2 (± 0.5)
Converged	32.7 (± 1.2)	63.5 (± 8.2)	23.7 (± 0.9)	73.8 (± 1.8)

We see that CIVILCOMMENTS sustain good performance using pretrained models at different convergence levels. FMoW and iWILDCAM on the other hand seems to have strong preference towards converged model, and the results worsen as the amount of data seen during training goes down. CAMELYON17 achieves the best performance when only 10% of data is seen, and the test accuracy decreases while training with models with higher level of convergence.

D.7 Ablation Studies on hyperparameters

α and ϵ We study the effect of changing Fish’s inner loop learning rate α and outer loop learning rate ϵ . To make the comparisons more meaningful, we keep $\alpha \cdot \epsilon$ constant while changing their respective values. See results in Figure D.3.

Meta steps N For most of the datasets we studied (all apart from CAMELYON17 where $T = 3$) we sample a N -sized subset of all T domains available for training (see Table D.11 for T of each dataset). Here we study when $N = 5, 10, 20$.

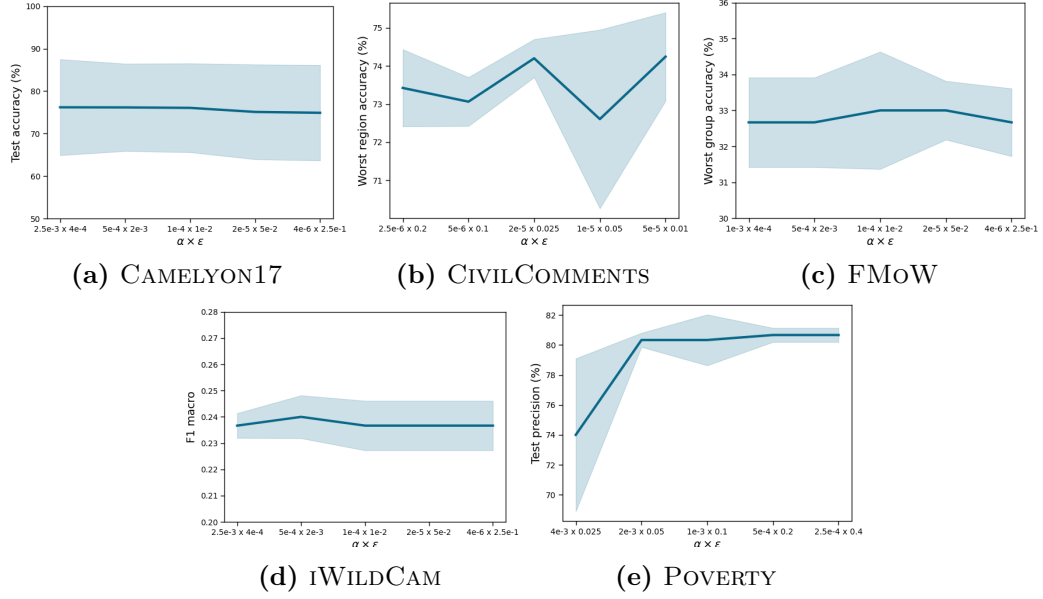


Figure D.3: Ablation studies on α and ϵ . Note that $\alpha \times \epsilon$ remains constant in all experiments, and the midpoint of each plot is the hyperparameter we chose to use to report our experiment results.

Table D.13: Ablation study on meta steps N .

N	FMoW	POVERTY	iWILDCAM	CIVILCOMMENTS
	Test Avg Acc	Test Pearson r	Test Macro F1	Test Worst Acc
5	33.0 (± 1.6)	80.3 (± 1.7)	23.7 (± 0.9)	74.3 (± 1.5)
10	32.7 (± 1.2)	80.0 (± 0.8)	23.7 (± 0.5)	73.4 (± 1.0)
20	33.3 (± 2.1)	77.7 (± 2.1)	23.7 (± 0.9)	72.6 (± 2.3)

In general altering these hyperparameters don't have a huge impact on the model's performance, however it does seem that when $N = 20$ the performance on some datasets (POVERTY and CIVILCOMMENTS) degrade slightly.

D.8 Tracking gradient inner product

In Figure D.4, we demonstrate the progression of inter-domain gradient inner products during training using different objectives. We train both **Fish** (blue) and **ERM** (yellow) until convergence while recording the normalized gradient inner products (i.e. cosine similarity) between minibatches from different domains used in each inner-loop. The gradient inner products are computed both before (dotted) and after (solid) the model's update. To ensure a fair comparison, we use

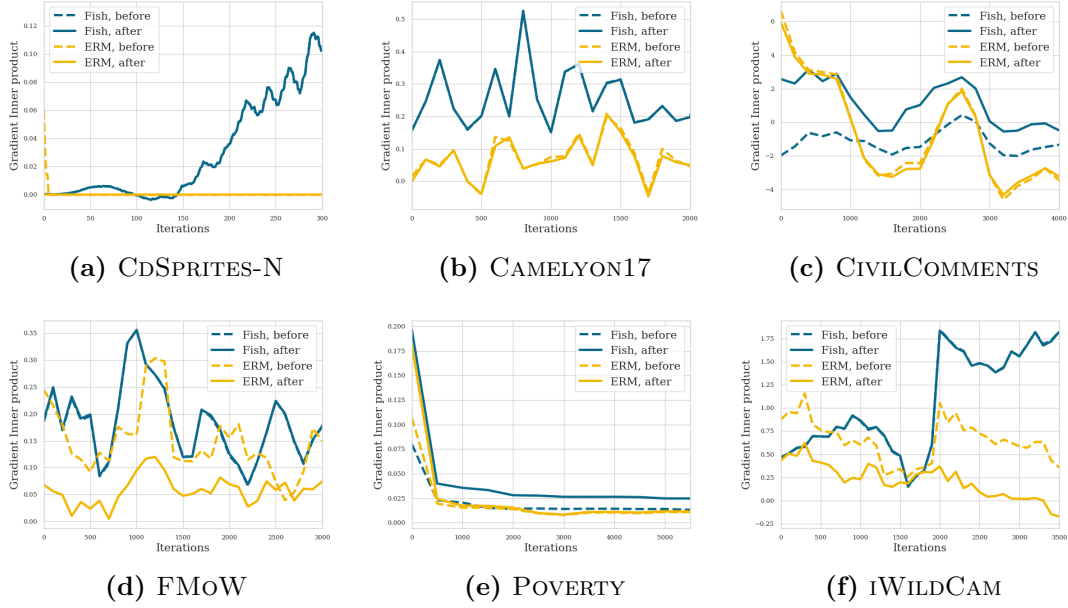


Figure D.4: Gradient inner product values during the training for CdSPRITES-N (N=15) and 5 different WILDS datasets.

the exact same sequence of data for Fish and ERM (see Appendix D.10 for more details).

Inevitably, the gradient inner product trends differently for each dataset since the data, types of domain splits and the choice of architecture are very different. In fact, the plot for CdSPRITES-N and POVERTY are significantly different from others, with a dip in gradient inner product at the beginning of training – this is because these are the two datasets that we train from scratch. On all other datasets, the gradient inner products are recorded when fine-tuning with Fish.

Despite their differences, there are some important commonalities between these plots: if we compare the pre-update (dotted) to post-update (solid) curves, we can see that ERM updates often result in the decrease of gradient inner product, while for Fish it can either increase significantly (Figure D.4c and Figure D.4e) or at least stay at the same level (Figure D.4a, Figure D.4b, Figure D.4d and Figure D.4f). As a result of this, we can see that the post-update gradient inner product of Fish is always greater than ERM at convergence.

The observations here shows that Fish is effective in increasing/maintaining the

level of inter-domain gradient inner product and sheds some lights on its efficiency on the datasets we studied.

D.9 Summary of DomainBed

DOMAINBED is a testbed for domain generalization that implements consistent experimental protocols across SOTA methods to ensure fair comparison. It contains 7 popular domain generalization datasets, including Colored MNIST (Arjovsky et al., 2019b), Rotated MNIST (Ghifary et al., 2015), VLCS (Fang et al., 2013), PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), Terra Incognita (Beery et al., 2018) and DomainNet (Peng et al., 2019), and offers comparison to a variety of SOTA domain generalization methods, including IRM (Arjovsky et al., 2019b), Group DRO (Hu et al., 2018a; Sagawa et al., 2019), Mixup (Yan et al., 2020), MLDG (Li et al., 2018a), Coral (Sun and Saenko, 2016b), MMD (Li et al., 2018b), DANN (Ganin et al., 2016c) and CDANN (Li et al., 2018).

D.10 Algorithm for tracking gradient inner product

To make sure that the gradients we record for ERM and Fish are comparable, we use the same sequence of S -minibatches to train both algorithms. See Algorithm 7 for de-

Algorithm 6: Function **GIP**

```

1: function GIP( $\{d_1, d_2, \dots, d_N\}, \theta$ ):
2:   for  $d_n \in \{d_1, d_2, \dots, d_N\}$  do
tails. 3:      $g_n = \partial l(d_n; \theta) / \partial \theta$ 
4:   end for
5:    $\bar{\mathbf{g}} = \frac{1}{S(S-1)} \sum_{i,j \in S}^{i \neq j} g_i \cdot g_j$ 
6:   return  $\bar{\mathbf{g}}$ 
```

D.11 t-SNE Visualisation of Learned Representations

In this section we visualize the representation learned by Fish and ERM with t-SNE (van der Maaten and Hinton, 2008) for PACS, VLCS and OfficeHome datasets. See Figure D.5.

Algorithm 7: Algorithm of collecting gradient inner product $\bar{\bar{g}}$ for **F**ish and **E**RM both **b**efore and **a**fter updates. See **G**IP in Algorithm 6.

```

1: Initialize Fish  $\theta_f \leftarrow \theta$ , ERM  $\theta_e \leftarrow \theta$ 
2: for  $i = 1, 2, \dots$  do
3:   //Get all minibatches
4:   for  $\mathcal{D}_n \in \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_N\}$  do
5:     Sample batch  $\mathbf{d}_n \sim \mathcal{D}_n$ 
6:   end for
7:   //GradInnerProd before update
8:    $\bar{\bar{g}}_{Fb} = \text{GIP}(\{d_1, d_2, \dots, d_N\}, \theta_f)$ 
9:    $\bar{\bar{g}}_{Eb} = \text{GIP}(\{d_1, d_2, \dots, d_N\}, \theta_e)$ 
10:  //Fish training
11:   $\tilde{\theta} \leftarrow \theta_f$ 
12:  for  $d_n \in \{d_1, d_2, \dots, d_N\}$  do
13:     $g_n = \partial l(\mathbf{d}_n; \tilde{\theta}) / \partial \tilde{\theta}$ 
14:    Update  $\tilde{\theta} \leftarrow \tilde{\theta} - \alpha g_n$ 
15:  end for
16:   $\theta_f \leftarrow \theta_f + \epsilon(\tilde{\theta} - \theta_f)$ 
17:  //Rearrange minibatches
18:   $d = \text{shuffle}(\text{concat}(d_1, d_2, \dots, d_N))$ 
19:   $\{\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_N\} = \text{split}(d)$ 
20:  //ERM training
21:  for  $\tilde{d}_n \in \{\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_N\}$  do
22:     $g_n = \partial l(\tilde{d}_n; \theta_e) / \partial \theta_e$ 
23:    Update  $\theta_e \leftarrow \theta_e - \alpha g_n$ 
24:  end for
25:  //GradInnerProd after update
26:   $\bar{\bar{g}}_{Fa} = \text{GIP}(\{d_1, d_2, \dots, d_N\}, \theta_f)$ 
27:   $\bar{\bar{g}}_{Ea} = \text{GIP}(\{d_1, d_2, \dots, d_N\}, \theta_e)$ 
28: end for
29: Return  $\bar{\bar{g}}_{Fb}, \bar{\bar{g}}_{Fa}, \bar{\bar{g}}_{Eb}, \bar{\bar{g}}_{Ea}$ 

```

We can see that Fish and ERM are both capable of forming distinctive label-clusters on PACS (first row) and VLCS (second row), however with ERM we can observe within each label cluster that sub-clusters of domains are forming. This is particularly the case for the red and yellow cluster in Figure D.5a and cyan cluster in Figure D.5c. The domain clustering phenomena is not observed for Fish. On the other hand, for the OfficeHome dataset where Fish outperforms ERM by more than 1%, we clearly see that Fish exhibits better label clustering performance than ERM.

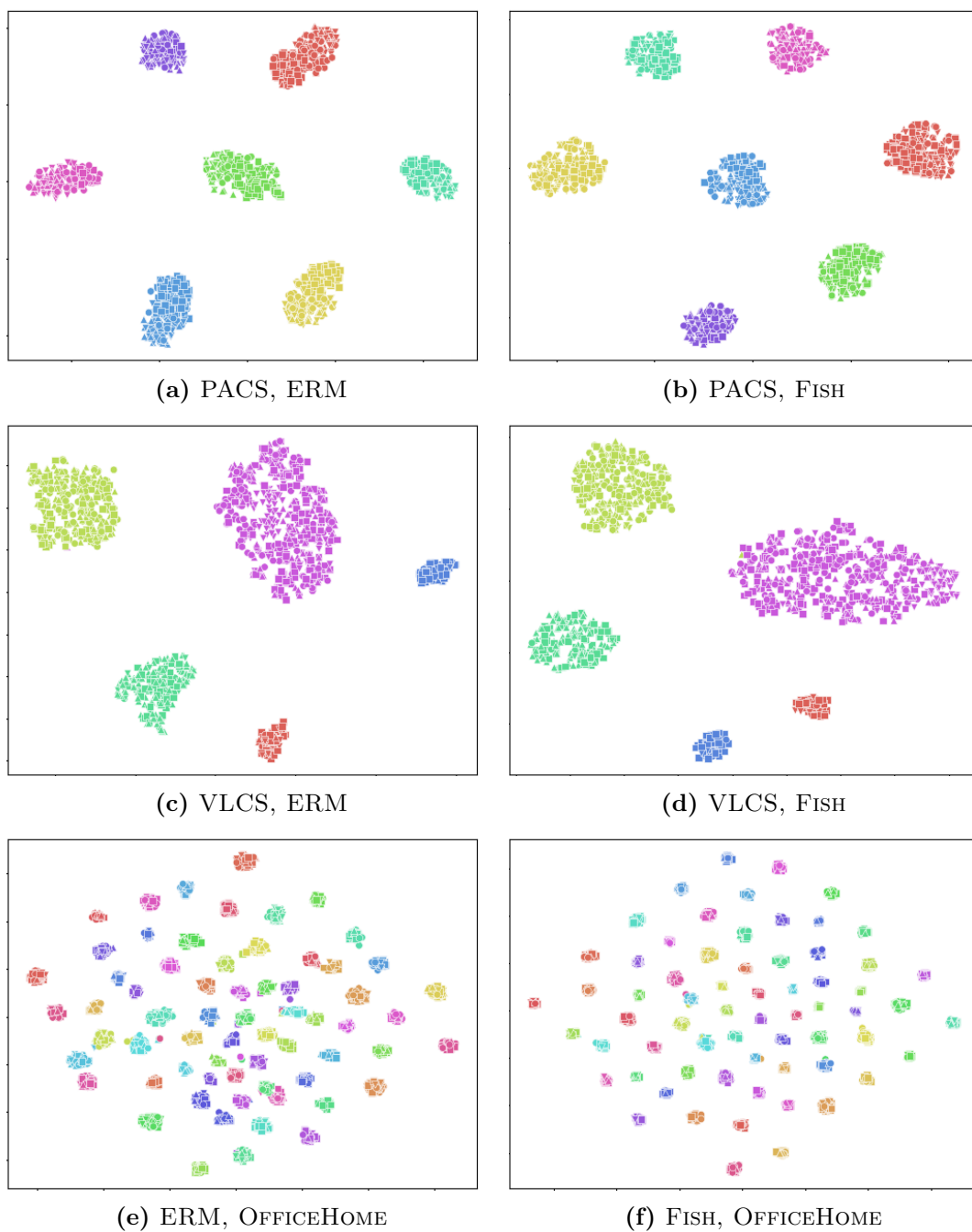


Figure D.5: t-SNE plot for PACS, VLCS and OfficeHome. Colors represent labels, markers (shape of each datapoint) represent domain.

E

Appendix: Adversarial Masking for Self-Supervised Learning

E.1 ADIOS Objectives

In this section we detail the objective used to optimise ADIOS+SimCLR, ADIOS+SimSiam and ADIOS+BYOL.

E.1.1 SimCLR

We show a simplified version of SimCLR architecture in Figure 6.4. In reality, the encoder $\mathcal{I}$ of SimCLR further factorises into two networks, including a base encoder $f(\cdot)$ which extracts the representation $\mathbf{h}$ used for downstream tasks, followed by a projection head $g(\cdot)$ which maps $\mathbf{h}$ to the final embedding that's used to compute the objective in Equation (6.4).

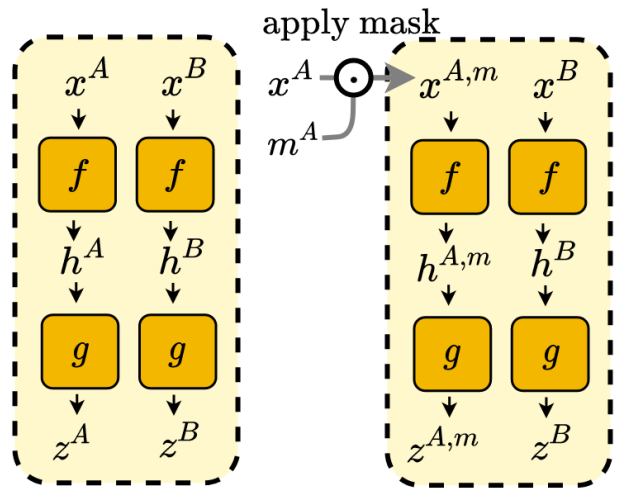


Figure E.1: *Left:* SimCLR. *Right:* SimCLR + ADIOS.

We visualise this in Figure E.1. It is helpful to establish SimCLR’s architecture as it lays the foundation for both SimSiam and BYOL.

E.1.2 SimSiam

Chen and He (2021) proposes SimSiam, an SSL method that can learn meaningful representations without negative examples. The forward pass of SimSiam also contains a base encoder and a projection head; however, different from SimCLR, for one of the augmentation streams the projection head is removed with a stop gradient operation applied to the base encoder (See Figure E.2). Authors find empirically that these two alterations are essential for preventing latent collapse in the absence of negative examples. The final objective of SimSiam is written as

$$\mathcal{L}_{\text{SimSiam}}(\mathbf{x}; \mathcal{I}) = \frac{1}{2} \left(\mathcal{D}(\mathbf{z}^A, \mathbf{h}^B) + \mathcal{D}(\mathbf{z}^B, \mathbf{h}^A) \right), \quad (\text{E.1})$$

where $\mathcal{D}$ denotes the negative cosine similarity, $\mathcal{I} = g \circ f$, and $\mathbf{z} = g(f(\mathbf{x}))$ while $\mathbf{h} = f(\mathbf{x})$. Note that the loss is the average of two distances due to the asymmetrical model.

Following the same intuition in Section 6.2, to adapt the objective for the ADIOS framework, we apply

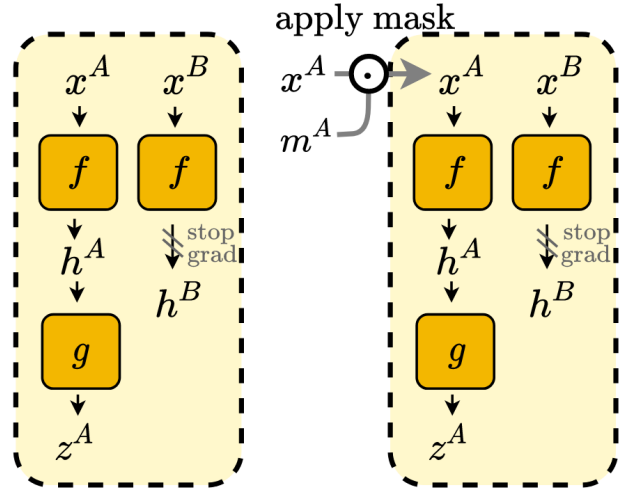


Figure E.2: *Left:* SimSiam. *Right:* SimSiam + ADIOS.

the masks learned by the occlusion model to one of the views. We can therefore arrive at our final objective,

$$\mathcal{L}_{\text{SimSiam}}^{\text{ADIOS}}(\mathbf{x}; \mathcal{I}, \mathcal{M}) = \frac{1}{2} \left(\mathcal{D}(\mathbf{z}^{A,m}, \mathbf{h}^B) + \mathcal{D}(\mathbf{z}^{B,m}, \mathbf{h}^A) \right), \quad (\text{E.2})$$

where $\mathbf{z}^{*,m} = g(f(\mathbf{x}^{*,m}))$.

E.1.3 BYOL

BYOL (Grill et al., 2020) is another SSL method that avoids the need of negative examples by performing an iterative online update. Similar to SimSiam, BYOL also adopts an assymetrical forward pass, however different from other approaches, the networks for two different augmentations do not share weights. See Figure E.3 for a visualisation.

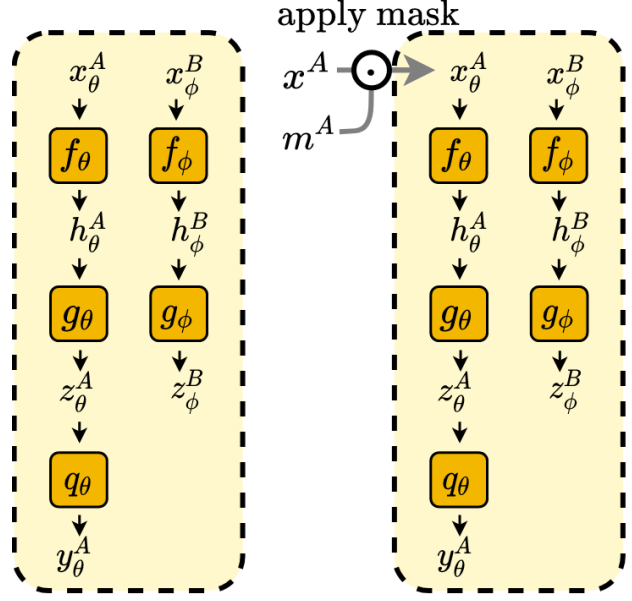


Figure E.3: *Left:* BYOL. *Right:* BYOL + ADIOS.

For the sake of clarity, we denote

the parametrisation of the two networks as θ and ϕ . The θ network is appended with an additional “predictor” q_θ , and is updated via gradient descent using the following objective, which is evaluated using the output of the θ network y_θ and output of the ϕ network z_ϕ

$$\mathcal{L}_{\text{BYOL}}(\mathbf{x}; \theta) = \frac{1}{2} \left(\mathcal{D}(\mathbf{y}_\theta^A, \mathbf{z}_\phi^B) + \mathcal{D}(\mathbf{y}_\theta^B, \mathbf{z}_\phi^A) \right), \quad (\text{E.3})$$

where $\mathcal{D}$ denotes the mean squared error. Again, the objective is the average between the two terms due to the assymetrical architecture.

On the other hand, ϕ is optimised using the following update rule

$$\phi \leftarrow \tau \phi + (1 - \tau) \phi, \quad (\text{E.4})$$

where $\tau \in [0, 1)$ controls the smoothness of the update.

To develop the ADIOS objective for BYOL, let us denote $\mathcal{I}$ as the composition of the two networks $\{\mathcal{I}_\theta, \mathcal{I}_\phi\}$. We can then write

$$\mathcal{L}_{\text{BYOL}}^{\text{ADIOS}}(\mathbf{x}; \mathcal{I}_\theta, \mathcal{M}) = \frac{1}{2} \left(\mathcal{D}(\mathbf{y}_\theta^{A,m}, \mathbf{z}_\phi^B) + \mathcal{D}(\mathbf{y}_\theta^{B,m}, \mathbf{z}_\phi^A) \right), \quad (\text{E.5})$$

where $\mathbf{y}_\theta^{*,m} = q_\theta(g_\theta(f_\theta(x^{*,m})))$. Both $\mathcal{I}_\theta$ and $\mathcal{M}$ are optimised through the min-max objective in Equation (6.6), whereas $\mathcal{I}_\phi$ is updated by Equation (E.4).

Down
3x3 conv. 8 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 8 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 16 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 16 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 16 stride 1 pad 1 & GroupNorm & ReLU
MLP
F.C. 128 & ReLU
F.C. 128 & ReLU
F.C. 256 & ReLU
Up
3x3 conv. 16 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 16 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 8 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 8 stride 1 pad 1 & GroupNorm & ReLU
3x3 conv. 8 stride 1 pad 1 & GroupNorm & ReLU
Occlusion Head
1x1 conv. N stride 1 pad 1 & SoftMax

Table E.1: U-Net Architecture.

E.2 Backbone of Occlusion Model

We use U-Net (Ronneberger et al., 2015) as the backbone of the occlusion model, which is commonly used for semantic segmentation. The model consists of a downsampling network, an MLP, and an upsampling network. We further apply an occlusion head layer with 1x1 kernel to map the output of U-Net to N masks. Refer to Table E.1 for the architecture we used for our experiments.

E.3 Setups of Classification Tasks

We develop our model using solo-learn (da Costa et al., 2021), a library for state-of-the-art self-supervised learning methods. For backbones we use ResNet-18 (He et al., 2016) and ViT-Tiny (Dosovitskiy et al., 2021) with patch size 16x16. Hyperparameters including optimiser, momentum, scheduler, epochs and batch size are shared across all models, as seen in Table E.2. We perform hyperparameter search on the learning rate of encoder for all models, and we include the optimal values used to generated the reported results in Table E.3; for the ADIOS models we also run search for the learning rate of the occlusion model, the penalty scaling λ and number of masks N . Refer to Table E.4 for the values used for these

Table E.2: Hyperparameters for pretraining, used for all models.

Name	Value
Optimiser	SGD
Momentum	0.9
Scheduler	warmup cosine
Epochs	500
Batch size	128

Table E.3: Learning rates for SimCLR, SimSiam and BYOL.

Architecture	Dataset	SimCLR	SimSiam	BYOL
ResNet18	ImageNet100-S	0.15	0.25	0.25
ResNet18	STL10	0.15	0.23	0.31
ViT-Tiny	ImageNet100-S	0.15	0.25	0.25
ViT-Tiny	STL10	0.15	0.11	0.23

parameters.

E.4 Setups of Transfer Learning

We fine-tune all the models using SGD with a momentum of 0.9 with cosine learning rate decay. Following protocol in Dosovitskiy et al. (2021), we use batch size 512 and no weight decay. We also run a small grid search on the learning rate with values including $\{0.001, 0.003, 0.01, 0.03\}$.

E.5 Setups of CLEVR

CLEVR (Johnson et al., 2017) is a dataset of rendered 3D objects. The dataset contains detailed attributes for each object, including shape, color, position, rotation, texture as well as mask, and is commonly used in visual question answering and multi-object representation learning. Utilising the rich annotations of CLEVR, we construct a challenging multi-label classification task, which we use to evaluate the quality of representations learned under different masking scheme.

For hyperparameters including optimiser, momentum, scheduler, epochs and batch size, we follow the same setup in Table E.2. We perform hyperparameter search on the learning rate of encoder and occlusion model, as well as the penalty scaling λ and number of masks N , which we list in Table E.5.

Table E.4: ADIOS hyperparameters for classification tasks.

	SimCLR+ADIOS	SimSiam+ADIOS	BYOL+ADIOS
<i>Dataset: ImageNet100-S, backbone: ResNet18</i>			
Enc. lr	0.13	0.85	0.24
Occ. lr	0.02	0.08	0.07
λ	0.57	0.29	0.40
N	4	4	4
<i>Dataset: ImageNet100-S, backbone: ViT-Tiny</i>			
Enc. lr	0.12	0.50	0.21
Occ. lr	0.03	0.07	0.33
λ	0.89	0.72	0.95
N	4	4	4
<i>Dataset: STL10, backbone: ResNet18</i>			
Enc. lr	0.21	0.52	0.49
Occ. lr	0.33	0.29	0.06
λ	0.29	0.79	0.72
N	6	6	6
<i>Dataset: STL10, backbone: ViT-Tiny</i>			
Enc. lr	0.14	0.56	0.29
Occ. lr	0.09	0.09	0.60
λ	0.50	0.12	0.18
N	6	6	6

Table E.5: Hyperparameters used for CLEVR.

	SimCLR	+ADIOS	SimSiam	+ADIOS	BYOL	+ADIOS
Enc. lr	0.2	0.3	0.7	0.5	0.5	0.4
Occ. lr	-	0.1	-	0.1	-	0.3
λ	-	0.2	-	0.8	-	0.9
N	-	4	-	4	-	4

E.6 CIFAR ResNet

As we mentioned, the authors of ResNet (He et al., 2016) proposes a slightly different architecture for CIFAR due to their small image size. The difference between the CIFAR-ResNet and standard ResNet lies in the first convolutional block (before `layer1`), which we provide a side by side comparison of in Table E.6 and Table E.7.

Table E.6: Standard ResNet.

7x7 conv. 64 stride 2 pad 3
BatchNorm
ReLU
3x3 MaxPool stride 2 pad 1

Table E.7: CIFAR ResNet.

3x3 conv. 64 stride 1 pad 2
BatchNorm
ReLU
-

As we see, the kernel size of the convolutional layer is smaller and the MaxPool

operation is removed to suit the small image size. We adopt this CIFAR-optimal ResNet for fine-tuning, but stick to the original architecture for linear evaluation to utilise the weights learned during pre-training, which resulted in the considerable underperformance on this metric for all 6 models.

F

Appendix: How Robust is Unsupervised Representation Learning to Distribution Shift?

F.1 Additional Experimental Results

F.1.1 CdSprites

In addition to our results in table A.2, where we use a dataset with perfectly correlated features ($r_{\text{id}} = 1$) to train the backbones, in fig. F.1 we vary r_{id} to analyse the effect of imperfectly correlated features. Notably, with imperfect correlation ($r_{\text{id}} < 1$), the OOD linear heads trained on top of the SL and SSL backbones perform perfectly. For the AE, we observe that the performance of the OOD linear head does not depend on the correlation r_{id} in the data used to train the backbones. Our results suggest that with imperfect correlation between features, SL and SSL models learn a linearly separable representation of the features, whereas AE does not.

In fig. F.2 we provide an ablation where we also vary r_{ood} , the correlation in the data used to train and evaluate the linear head. Figures F.2b and F.2c corroborate our results that SSL performs on par with SL for $r_{\text{id}} < 1$ and strictly better when

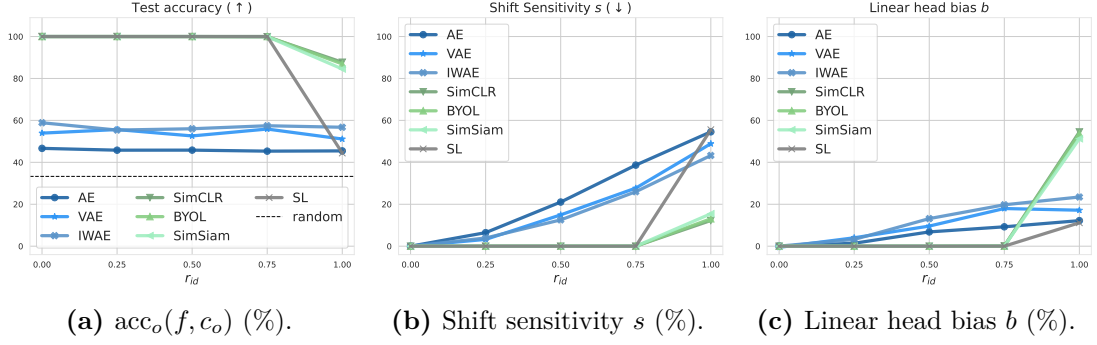


Figure F.1: Evaluations on the CdSprites dataset with $r_{id} \in \{0.25, 0.5, 0.75, 1.0\}$. We report shape classification accuracy using OOD-trained linear head ($acc_o(f, c_o)$), shift sensitivity s , and linear head bias b . Results are shown for *individual* models from the class of AE (blue), SSL (green), and SL (grey) algorithms. The black horizontal line denotes the random baseline (33.3% for three classes).

$r_{id} = 1$. For the AE (Figure F.2a), we observe an interesting pattern where the performance of the OOD linear head depends on the OOD correlation r_{ood} , but not on the correlation r_{id} in the data used to train the backbones. Hence, the ablation corroborates our result that SL and SSL models learn a linearly separable representation of the shape and color features when there is an imperfect correlation between the features, whereas AE does not.

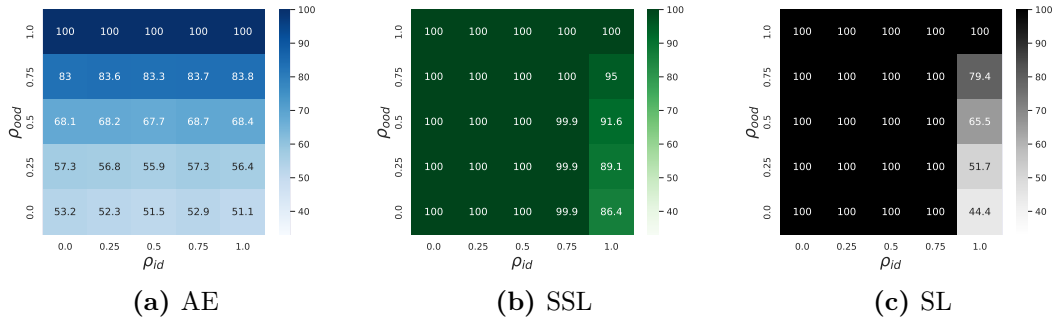


Figure F.2: Correlation coefficient ablation for CdSprites. Shape classification accuracy for the CdSprites experiment with varying correlation of the ID training data (r_{id} , x-axis) and OOD training and test data (r_{ood} , y-axis). Backbones were trained on data with correlation r_{id} and linear classifiers trained and evaluated on top of the frozen backbones with correlation r_{ood} .

F.2 Architecture and Hyperparameters

In this appendix we list the architecture and hyperparameters used in our experiments. Our code is developed on the amazing `solo-learn` code base (da Costa

Table F.1: Hyperparameter search range for MNIST-CIFAR, including base channel size of CNN (C), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr. scheduler$).

	C	$lr.$	$wd.$	Optim.	$lr. scheduler$
AE	{16, 32, 64, 128}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SSL	{16, 32, 64, 128}	uniformly sampled from [0.1, 1]	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SL	{16, 32, 64, 128}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}

et al., 2022), which is originally developed as a library for SSL algorithms. For all experiments we follow the standard set of augmentations established in He et al. (2020a); Chen et al. (2020c), including random resize crop, random color jittering, random grayscale, random Gaussian blur, random solarisation and random horizontal flip. An exception is the CdSprites experiment where we remove color jittering, as color classification is one of the tasks we are interested in and color jittering would add noise to the labels. For MNIST-CIFAR, we independently apply random augmentation to MNIST and CIFAR respectively (drawn from the same set of augmentations as detailed above) and then concatenate them to construct training examples.

Please see implementation details for each dataset in the respective subsection.

F.2.1 MNIST-CIFAR

We use the same hyperparameter search range for models in each category of AE, SSL and SL, as outlined in table F.1. The chosen hyperparameters for each model are specified in table F.2.

In Shah et al. (2020) where MNIST-CIFAR was originally proposed, authors utilised more complex backbone architecture such as DenseNet and MobileNet. However in our experiments, we find that a lightweight 4-layer CNN can already achieve very high accuracy on both MNIST and CIFAR. The architecture of the CNN we use can be found in table F.3. Note that for SL and SSL we only use the encoder and for AE we use the decoder as well. The size of base channel C and latent dimension L are found through hyperparameter search.

Table F.2: Chosen hyperparameters for MNIST-CIFAR including latent dimension (L), base feature size of CNN (C), batch size (B), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr.scheduler$).

	L	C	B	lr.	wd.	Optim.	lr. scheduler
AE	128	16	128	1e-3	0	Adam	warmup cosine
VAE	128	32	128	1e-4	0	Adam	warmup cosine
IWAE	128	32	128	1e-4	0	Adam	step
β -VAE	128	16	128	1e-4	0	Adam	step
SimCLR	128	32	128	6e-1	1e-4	SGD	warmup cosine
BYOL	128	64	128	7e-1	0	SGD	warmup cosine
SimSiam	128	128	128	6e-1	1e-5	SGD	warmup cosine
Supervised	128	16	128	1e-4	0	SGD	warmup cosine

Encoder
Input $\in \mathbb{R}^{3 \times 64 \times 32}$
4x4 conv. C stride 2x2 pad 1x1 & ReLU
4x4 conv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. $4C$ stride 2x2 pad 1x1 & ReLU
4x1 conv. $4C$ stride 2x1 pad 1x0 & ReLU
4x4 conv. L stride 1 pad 0, 4x4 conv. L stride 1x1 pad 0x0
Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. $4C$ stride 1x1 pad 0x0 & ReLU
4x1 upconv. $4C$ stride 2x1 pad 1x0 & ReLU
4x4 upconv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1 & Sigmoid

Table F.3: CNN architecture, MNIST-CIFAR dataset.

F.2.2 CdSprites

We found all models to be relatively robust to hyperparameters, as most configurations result in close to perfect shape and color classification accuracy on the ID validation set. The chosen hyperparameters for each model are specified in table F.4. We omit β -VAE from the comparison, as we empirically found that $\beta = 1$ leads to the best performance on the ID validation set and therefore the results for the β -VAE would be similar to the VAE. We use the same augmentations (random crops and horizontal flips) for all models and use no color augmentations in order to keep the invariance of the learned representations with respect to color. The encoder and decoder architectures are described in table F.5.

Table F.6: Hyperparameter search range for Camelyon17, including decoder type, latent dimension (L), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr. scheduler$).

	Decoder type	L	$lr.$	$wd.$	Optim.	$lr. scheduler$
AE	[CNN, MLP, ResNet]	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}
SSL	-	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1, 1}	{0, 1e-3, 1e-4, 1e-5}	{Adam, SGD}	{warmup cosine, step, none}
SL	-	{256, 512, 1024}	{1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 1e-1, 5e-1}	{0, 1e-4}	{Adam, SGD}	{warmup cosine, step, none}

Table F.4: Chosen hyperparameters for CdSprites including latent dimension (L), base feature size of CNN (C), batch size (B), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr.schedulr$).

	L	C	B	lr.	wd.	Optim.	none
AE	512	64	128	5e-5	1e-4	Adam	none
VAE	512	64	128	5e-5	1e-4	Adam	none
IWAE	512	64	128	5e-5	1e-4	Adam	none
SimCLR	64	32	64	5e-3	1e-5	SGD	warmup cosine
BYOL	64	32	64	5e-1	1e-5	SGD	warmup cosine
SimSiam	64	32	64	8e-2	1e-5	SGD	warmup cosine
Supervised	512	64	128	5e-5	1e-4	Adam	none

Encoder

Input $\in \mathbb{R}^{3 \times 64 \times 64}$
4x4 conv. C stride 2x2 pad 1x1 & ReLU
4x4 conv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. $8C$ stride 2x2 pad 1x1 & ReLU
4x4 conv. L stride 1 pad 0

Decoder

Input $\in \mathbb{R}^L$
4x4 upconv. $8C$ stride 1x1 pad 0x0 & ReLU
4x4 upconv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1

Table F.5: CNN architecture, CdSprites dataset.

F.2.3 Camelyon17 and FMoW

For hyperparameters including batch size, max epoch and model selection criteria, we follow the same protocol as in WILDS (Koh et al., 2021): for Camelyon17 we use a batch size of 32, train all models for 10 epochs and select the model that results in the highest accuracy on the validation set, and for FMoW the batch size is 32, max epoch is 60 and model selection criteria is worst group accuracy on OOD validation set. For the rest, we use the same hyperparameter search range for models in each

category of AE, SSL and SL, as outlined in table F.6. The chosen hyperparameters for Camelyon17 are specified in table F.7, and for FMoW in table F.8. For Camelyon17-C and FMoW-C we use these same hyperparameters.

Table F.7: Chosen hyperparameters for Camelyon17 including latent dimension (L), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr. scheduler$).

	Decoder	lr.	wd.	Optim.	lr. scheduler
AE	ResNet	5e-4	1e-5	SGD	warmup cosine
VAE	MLP	1e-4	0	Adam	none
IWAE	MLP	1e-4	0	Adam	none
β -VAE	MLP	1e-4	0	Adam	none
SimCLR	-	1e-1	0	SGD	none
BYOL	-	1e-1	1e-5	SGD	warmup cosine
SimSiam	-	1e-1	1e-5	SGD	warmup cosine
Supervised	-	1e-3	1e-3	SGD	none

Table F.8: Chosen hyperparameters for FMoW including latent dimension (L), learning rate ($lr.$), weight decay ($wd.$), optimiser ($optim.$), learning rate scheduler ($lr. scheduler$).

	Decoder	lr.	wd.	Optim.	lr. scheduler
AE	CNN	1e-1	1e-4	SGD	none
VAE	MLP	1e-6	1e-4	Adam	step
IWAE	MLP	1e-6	1e-4	Adam	step
β -VAE	MLP	1e-6	1e-4	Adam	step
SimCLR	-	5e-4	1e-3	SGD	step
BYOL	-	1e-2	1e-4	SGD	step
SimSiam	-	5e-4	0	SGD	step
Supervised	-	1e-4	0	Adam	step

We follow Koh et al. (2021) and use DenseNet121 (Huang et al., 2017) as backbone architecture. For the decoder of the AE models, we perform hyperparameter search between three architectures: a CNN (see table F.9), a simple 3-layer MLP (see table F.10) and a ResNet-like decoder with skip connections (see table F.11).

CNN, Decoder
Input $\in \mathbb{R}^L$
4x4 upconv. $8C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $8C$ stride 2x2 pad 0x0 & ReLU
4x4 upconv. $4C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. $2C$ stride 2x2 pad 1x1 & ReLU
4x4 upconv. C stride 2x2 pad 1x1 & ReLU
4x4 upconv. 3 stride 2x2 pad 1x1 & Sigmoid

Table F.9: CNN architecture, Camelyon17 dataset.

MLP, Decoder
Input $\in \mathbb{R}^L$
fc. $2L$ & ReLU
fc. $4L$ & ReLU
fc. $3*96*96$ & ReLU

Table F.10: MLP architecture, Camelyon17 dataset.

ResNet, Decoder
Input $\in \mathbb{R}^L$
fc. 2048 & ReLU
3x3 conv. $16C$ stride 1x1 pad 1x1
3x3 conv. $16C$ stride 1x1 pad 1x1
x2 upsample
3x3 conv. $8C$ stride 1x1 pad 1x1
3x3 conv. $8C$ stride 1x1 pad 1x1
x2 upsample
3x3 conv. $8C$ stride 1x1 pad 1x1
3x3 conv. $8C$ stride 1x1 pad 1x1
3x3 conv. 3 stride 1x1 pad 1x1

Table F.11: ResNet decoder architecture, Camelyon17 dataset.

F.3 Constructing Camelyon17-CS and FMoW-CS

We subsample Camelyon17 and FMoW dataset to create varying degree of spurious correlation between the domain and label information. We refer to these datasets as Camelyon17-CS and FMoW-CS. To construct such datasets, we first find some domain-label pairing in each dataset, such that if we sample the dataset according to this pairing, the population of each class with respect to the total number of examples in the dataset remains relatively stable. The $r_{\text{id}} = 1$ versions of both Camelyon17-CS and FMoW-CS can be acquired by simply subsampling the dataset following the domain label pairing; to ensure fairness in comparison, when constructing the $r_{\text{id}} \in \{0, 0.5\}$ versions of these datasets, we first mix in anti-bias samples (i.e. samples that are not in the domain-label pairing) to change the spurious correlation, and then subsample the dataset such that the size of the dataset is the same as the $r_{\text{id}} = 1$ version.

The domain-label pairing of Camelyon17-CS can be found in table F.12 and FMoW-CS in table F.13.

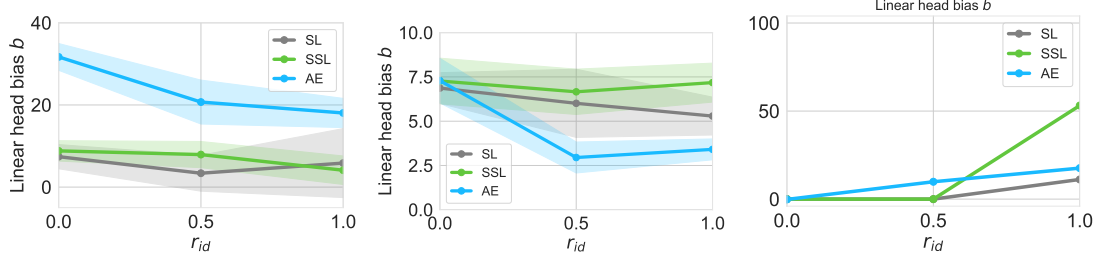


Figure F.3: Linear head bias on controllable shift datasets

Linear head bias We also plot the linear head bias for the experiments conducted on Camelyon17-CS, FMoW-CS, and CdSprites in Figure F.3. The experimental protocol follows that from Figure A.3.

Table F.12: Domain-label pairing for Camelyon17-CS.

Domain (hospital)	Label
Hospital 1, 2	Benign
Hospital 3	Malignant

Table F.13: Domain-label pairing for FMoW-CS.

Domain (region)	Label
Asia	Military facility, multi-unit residential, tunnel opening, wind farm, toll booth, road bridge, oil or gas facility, helipad, nuclear powerplant, police station, port
Europe	Smokestack, barn, waste disposal, hospital, water treatment facility, amusement park, fire station, fountain, construction site, shipyard, solar farm, space facility
Africa	Place of worship, crop field, dam, tower, runway, airport, electric substation, flooded road, border checkpoint, prison, archaeological site, factory or powerplant, impoverished settlement, lake or pond
Americas	Recreational facility, swimming pool, educational institution, stadium, golf course, office building, interchange, car dealership, railway bridge, storage tank, surface mine, zoo
Oceania	Single-unit residential, parking lot or garage, race track, park, ground transportation station, shopping mall, airport terminal, airport hangar, lighthouse, gas station, aquaculture, burial site, debris or rubble

F.4 Other Related Work

Explicit, extreme distribution shift Refers to when the features that caused distribution shift is explicit, known, controllable, and in some cases, extreme (e.g. MNIST-CIFAR, CdSprites). This type of settings are popular in works that investigate simplicity bias (Shah et al., 2020), dataset bias (Torralla and Efros,

2011a) and shortcut learning (Geirhos et al., 2020; Lapuschkin et al., 2019), as it allows for users to easily adjust the level of distribution shift between train and test set. Various specialised methods that either mitigate or address these problems under the supervised learning regime have been proposed, including Teney et al. (2022) that proposes to find shortcut solutions by ensembles, Luo et al. (2021) that avoids shortcut learning by extracting foreground objects for representation learning only, as well as Torralba and Efros (2011a); Kim et al. (2019); Le Bras et al. (2020) that re-sample the dataset to reduce the spurious correlation.

Importantly Kirichenko et al. (2022); Kang et al. (2020) shows that this extreme simplicity bias can be mitigated in some cases by retraining the final linear layer. This is a game changer, as it for the first time decouples the bias of the linear head from that of the main representation learning model. Interestingly, Robinson et al. (2021) also investigates the robustness of contrastive-SSL methods against simplicity bias. However, without training the linear head on OOD data, their finding is opposite to ours — that SSL methods are not able to avoid shortcut solutions.

Implicit, subtle distribution shift This type of problem is commonly seen in realistic distribution shift datasets (such as WILDS) and are often studied in Domain generalisation (DG). In this regime, the training data are sampled from multiple domains, while data from a new, unseen target domain is used as test set. Note that here we omitted the discussion on domain adaptation (DA), as in DA models typically have access to unlabelled data from the target domain, which is different from the settings we consider in this work.

There are mainly two lines of work in DG, namely 1) *Distributional Robustness approaches (DRO)*, which minimises the worst group accuracy to address covariate shift (Gretton et al., 2009a,b) and subpopulation shift (Sagawa et al., 2020; Hu et al., 2018b); 2) *Domain invariance*, which consists of methods that directly learn representations that are invariant across domains (Ben-David et al., 2010; Ganin et al., 2016b; Wang et al., 2019), encourage the alignment of gradients

from different domains (Koyama and Yamaguchi, 2021b; Lopez-Paz and Ranzato, 2017; Shi et al., 2022b), or optimise for representations that result in the same optimal classifier for different domains (Arjovsky et al., 2019a). Apart from these supervised learning methods, the recent advancement of SSL has also inspired works in unsupervised domain generalisation (Zhang et al., 2022; Harary et al., 2022). While all these methods achieved impressive performance, we note that they are all specially designed for DG with the majority of the methods relying on domain information and label information. In contrast, our work studies how existing standard representation learning methods such as SSL and AE performs on DG tasks, with none of the methods relying on human annotations.