Natural inductive biases for artificial intelligence

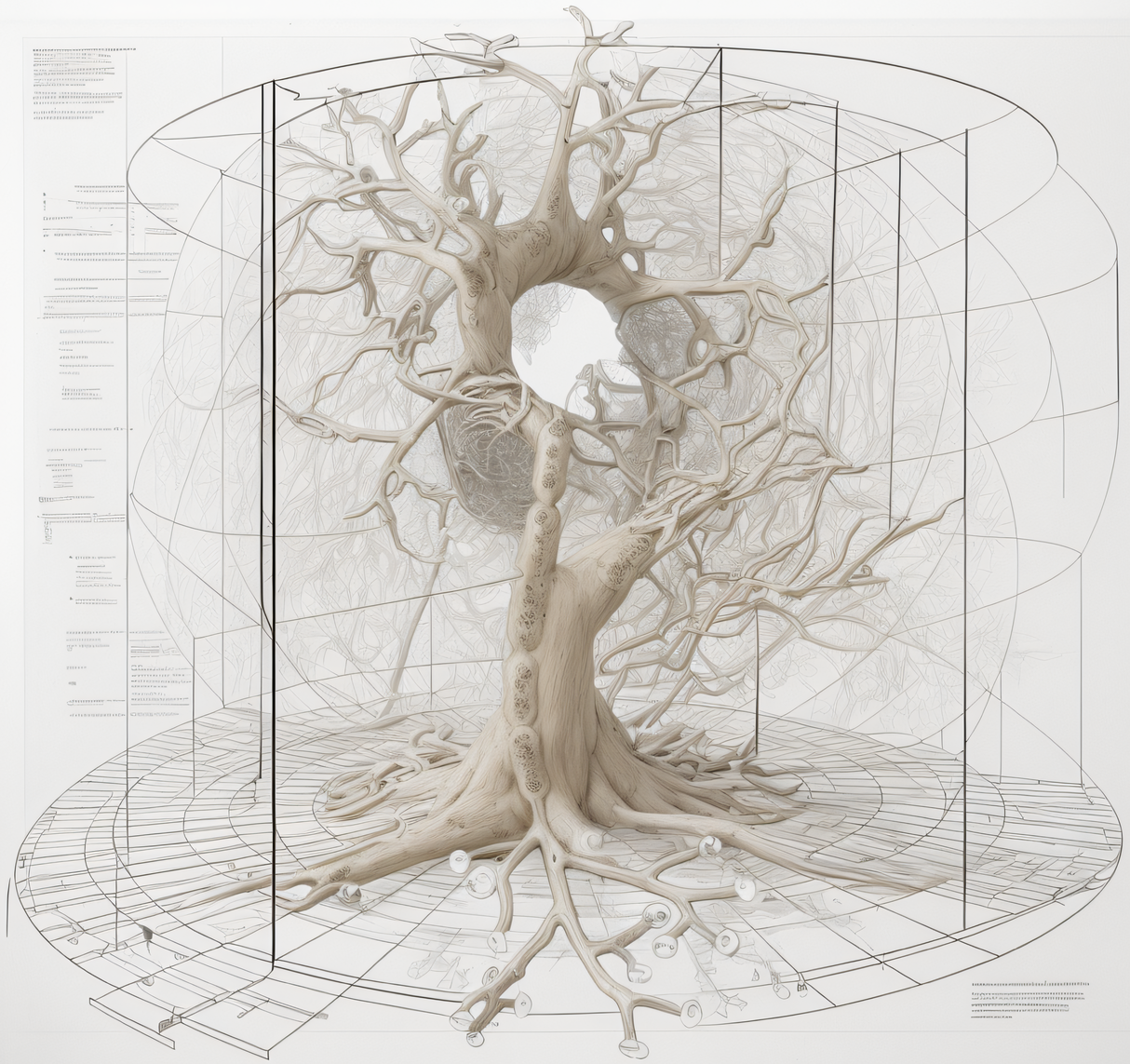
Keller, T.A.

Publication date
2023
Document Version
Final published version

Natural Inductive Biases for Artificial Intelligence

T. Anderson Keller

Natural Inductive Biases for Artificial Intelligence



T. Anderson Keller

Natural Inductive Biases for Artificial Intelligence

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 7 november 2023, te 16.00 uur

door Thomas Anderson Keller
geboren te California

Promotiecommissie

Promotor:	prof. dr. M. Welling	Universiteit van Amsterdam
Copromotor:	dr. H.C. van Hoof	Universiteit van Amsterdam
Overige leden:	prof. dr. C.M.A. Pennartz	Universiteit van Amsterdam
	prof. dr. S.M. Bohte	Universiteit van Amsterdam
	prof. dr. B.A. Olshausen	UC Berkeley
	dr. K.B. Dobs	Justus-Liebig University Giessen
	dr. ir. E.J. Bekkers	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

ACKNOWLEDGEMENTS

To my supervisor Max, I express my deepest gratitude. Over the past years, I have learned from you not just how to be a scientist, but what it means to be truly excited by research. I can recall multiple meetings where I entered despondent, and left energized by the way you passionately described the elegant beauty of conceptual ideas. Your enthusiasm has inspired me in the truest sense – I will be eternally grateful for this. I hope one day to be able to share this same excitement with my students and the world.

While this dissertation is mainly a reflection of work completed during the past years of my PhD, it is abundantly clear that the mere opportunity for this work to begin would not have existed had it not been for the undying love and support I have received every day from my parents. There are not words that can express the gratitude I feel for the foundation upon which they raised me, the wisdom and virtues they instilled in me, and the demonstration of what it truly means to love unconditionally. Thank you mom and dad, I dedicate this work to you.

Throughout the good and the bad, my family and friends have been there for me, and I therefore express my sincere appreciation to them. Without their support, there is certainly no way I would have remained sane long enough to complete this work. Adam, thank you for being the epitome of a brother; I know I can count on you for anything, especially for a laugh to cheer me up, you helped me far more than you know. To my friends Victor, Emiel, and Jorn, from the moment you welcomed me into the lab on the first day, your friendship has been the bedrock of my time here in Amsterdam, and I honestly do not believe I would have made it through without you. Victor, thank you specifically for all the regaettón and jamón; Emiel, thank you for all the long walks, discussions and life advice; and Jorn, thank you for the food, kindness, and friendship. I greatly admire all of you both professionally and personally, and I feel incredibly fortunate to have gotten to opportunity to learn from you over the past years. To my friends Daniel, Karen, Bas, Teddy, and Benedetta, thank you for sharing your time with me over the past years, for indulging me in conversations, and for showing me how to relax and enjoy life.

To those who provided me with guidance: Patrick Forré, Herke van Hoof, Eric Nalisnick, Erik Bekkers, Jan-Willem van de Meent, Sara Maglicane, and Zeynep Akata; thank you for always being available for discussion and for your belief in

me. Your guidance was invaluable throughout my PhD. To the members of the lab in Amsterdam: Artem Moskalev, Thomas Kipf, Metod Jazbec, Ben Miller, Marco Federici, Heiko Zimmermann, Tim Bakker, Babak Esmaeili, David Ruhe, Priyank Jaini, Elise van der Pol, Sadaf Gulshad, Fiona Lippert, Ivan Sosnovik, Putri van der Linden, Jakub Tomczak, Maximilian Ilse, Yue Song; it has been my honor to get to work with and learn from all of you, thank you. To my students Qinghe Gao, Samarth Bhargav, Noah van Grinsven, and Fiorella Wever; thank you for the discussions and collaborations. I have likely learned more from you than you have from me, and I am truly excited to see how your careers progress. Finally, thanks to Ninon Lizé Masclef for the design of the cover of this book and engaging discussions.

PUBLISHED CONTENT AND CONTRIBUTIONS

This thesis is based on the following publications:

Keller, T. Anderson, Qinghe Gao, and Max Welling (2021). "Modeling Category-Selective Cortical Regions with Topographic Variational Autoencoders". In: *SVRHM 2021 Workshop @ NeurIPS*.

Keller, T. Anderson and Max Welling (2021a). "Predictive Coding With Topographic Variational Autoencoders". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.

Keller, T. Anderson and Max Welling (2021b). "Topographic VAEs learn Equivariant Capsules". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc.

Keller, T. Anderson, Lyle Muller, Terrence Sejnowski, and Max Welling (2023). *Traveling Waves Encode the Recent Past and Enhance Sequence Learning*. Under Review. arXiv: 2309.08045 [cs.NE].

Keller, T. Anderson, Xavier Suau, and Luca Zappella (Nov. 2023). "Homomorphic Self-Supervised Learning". In: *Transactions on Machine Learning Research*.

Keller, T. Anderson and Max Welling (2023). "Neural Wave Machines: Learning Spatiotemporally Structured Representations with Locally Coupled Oscillatory Recurrent Neural Networks." In: *Proceedings of the 40th International Conference on Machine Learning*. Proceedings of Machine Learning Research. PMLR.

Song, Yue, T. Anderson Keller, and Max Welling (2023). "Flow Factorized Representation Learning". In: *Advances in Neural Information Processing Systems*. Vol. 37. Curran Associates, Inc.

For all of the above publications with T. Anderson Keller as first author, he contributed to the majority of the formulation of the methodology, performed the experiments, and writing of the paper. For the paper with Yue Song as first author, T. Anderson Keller contributed to the conception of the idea, formulation of the methodology, and writing the paper, but did not perform the experiments. Permission from Yue Song was obtained to include this publication in the thesis. For all publications above Max Welling provided indispensable guidance, insight, and supervision.

TABLE OF CONTENTS

SUMMARY

The study of inductive bias is one of the most all encompassing in all of machine learning. Inductive biases define not only the efficiency and speed of learning, but also what is ultimately possible to learn by a given machine learning system. The history of modern machine learning is intertwined with that of psychology, cognitive science and neuroscience, and therefore many of the most impactful inductive biases have come directly from these fields. Examples include convolutional neural networks, stemming from the observed organization of natural visual systems, and artificial neural networks themselves intending to model idealized abstract neural circuits. Given the dramatic successes of machine learning in recent years however, more emphasis has been placed on the engineering challenges faced by scaling up machine learning systems, with less focus on their inductive biases. This thesis will be an attempted step in the reverse direction. To do so, we will cover both naturally relevant learning algorithms, as well as natural structure inherent to neural representations. We will build artificial systems which are modeled after these natural properties, and we will demonstrate how they are both beneficial to computation, and may serve to help us better understand natural intelligence itself.

SAMENVATTING

De studie van inductieve bias is een van de meest allesomvattende binnen heel machine learning. Inductieve vooroordelen bepalen niet alleen de efficiëntie en snelheid van leren, maar ook wat uiteindelijk mogelijk is om te leren door een gegeven machine learning systeem. De geschiedenis van modern machine learning is verweven met die van psychologie, cognitieve wetenschap en neurowetenschap, en daarom zijn veel van de meest impactvolle inductieve vooroordelen rechtstreeks afkomstig uit deze velden. Voorbeelden zijn convolutionele neurale netwerken, voortkomend uit de waargenomen organisatie van natuurlijke visuele systemen, en kunstmatige neurale netwerken die bedoeld zijn om geïdealiseerde abstracte neurale circuits te modelleren. Gezien de dramatische successen van machine learning in recente jaren is er echter meer nadruk gelegd op de technische uitdagingen die gepaard gaan met het opschalen van machine learning systemen, met minder focus op hun inductieve vooroordelen. Deze thesis zal een poging zijn om een stap in de tegenovergestelde richting te zetten. Daartoe zullen we zowel natuurlijk relevante leer algoritmes bespreken, als ook de natuurlijke structuur inherent aan neurale representaties. We zullen kunstmatige systemen bouwen die gemodelleerd zijn naar deze natuurlijke eigenschappen, en we zullen aantonen hoe ze zowel voordelig zijn voor de berekening, als ons kunnen helpen om natuurlijke intelligentie beter te begrijpen.

QUOTES

We are but whirlpools in a river of ever-flowing water. We are not stuff that abides, but patterns that perpetuate themselves.

Norbert Wiener, 1950

I suspect that a deeper mathematical study of the nervous system ... may alter the way in which we look at mathematics and logics proper.

John von Newmann, 1958

If we could make a neural network model which has the same capability for pattern recognition as a human being, it would give us a powerful clue to the understanding of the neural mechanism in the brain.

Kunihiko Fukushima, 1980

Neurons have all kinds of components, or properties, to them and in evolutionary biology, if you have some little quirk in how a molecule works or how a cell works, evolution will sharpen it up and make it into a useful feature rather than a glitch.

John Hopfield, 2020

Max, I've figured out how the brain works.

Geoffrey Hinton, 2001

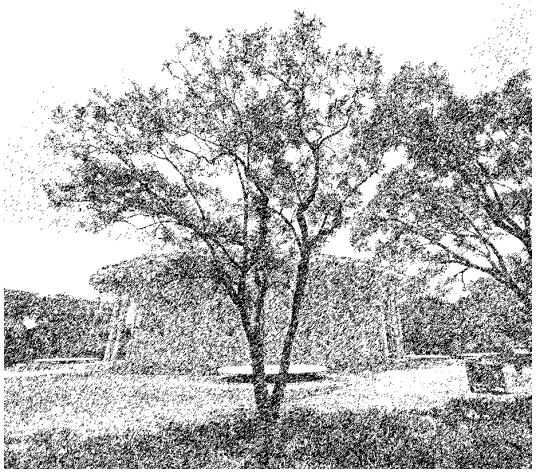
Andy, I've figured out how the brain works.

Max Welling, 2019

PREFACE

It was one in the morning as I walked in a trance through the dimly lit campus gardens, back towards my dormitory. In passing, my gaze drifted from the base of an underlit tree to its branches, and my attention fixated on the bright underside of a single leaf. As the distance between us shrank, the leaf transformed before my eyes, slowly revealing it's concave inner surface to me. In an instant I'll never forget, my awareness broadened, and I saw not just one leaf transforming, but all leaves of the tree shifting in unison, like a flock of starlings performing a coordinated dance through the night sky. As each leaf traced out its twisted path, their distinct motions were somehow jointly harmonious and intuitive, as if I knew the exact position of each leaf an instant before ever seeing it.

I spent an unhealthy amount of time staring at that tree. For first time not simply stunned by the beauty of the nature before me, but rather by of the beauty of how I was processing it. I was astounded by the mind's ability to register and convey the seemingly infinite dimensional symmetric structure of the outside world to my seemingly slow unitary consciousness. This moment has stuck with me, and since then, I have learned to reconsider how we understand objects not just on their own, but as compositions of parts, geometric relationships, and transformations. This thesis represents my journey studying the computational mechanisms by which the brain could represent these structures and relationships, and I am coming to find the insights along the way are almost as beautiful as the tree itself that night. My hope is that this work shares a piece of that beauty with you.



Preliminaries

Chapter 1

INTRODUCTION

In the silent darkness behind every pair of eyes, a chorus of electrical signals intertwines, harmonizes, and cascades through a web of connectivity, orchestrating the symphony of beliefs and behaviors we call intelligence. This concert reverberates across species from humans to apes, vertebrates to cephalopods, insects to a myriad of other animals. It is the unseen conductor behind gymnast's gravity-defying ballet, the invisible cartographer guiding the monarch butterfly on its great migration, the composer to the whale pod's underwater melody, and the endless source of inspiration for the philosopher and neuroscientist. The complex systems which engender these abilities have been carefully sculpted by environmental pressures over millions of years to suit their unique environmental needs and are highly optimized for their respective functions. For centuries, scientists have studied such systems with profound admiration, hoping to gain a greater understanding of our own natural intelligence, and potentially the beautiful truth underlying abstract intelligence as a whole. In the following we will delve into this history, the fundamentals of learning systems, and why we believe there is still much to be gained from studying the successes of nature in this regard. This introduction is intended for a general audience, aiming to refuel the cross-disciplinary dialogue that has historically nurtured the evolution of artificial intelligence.

1.1. The Entwined Tale of Natural and Artificial Intelligence

From the earliest days of computation, the development of computers has been inextricably intertwined with that of artificial intelligence. Early pioneers of digital computers took significant inspiration from natural intelligence, and the echoes of this natural inspiration can still be found throughout today's most advanced technologies.

In 1943, neurophysiologist Warren McCulloch and logician Walter Pitts published a new theory proposing that a network of simple neurons may be able to perform logical operations (McCulloch and Pitts, 1943). Using what they termed 'a logical calculus', they showed that a network of abstract 'all-or-nothing' neurons is able to express any logical expression under certain conditions, and that equivalently every

network can be described in such logical terms. Just over a year later, John von Neumann published the First Draft of a Report on the EDVAC (Neumann, 1945) (the Electronic Discrete Variable Automatic Computer) now widely agreed to be the first description of a modern digital computer, setting the stage for the digital revolution of the 20th century. In that report, von Neumann cites only a single other work, precisely that of McCulloch and Pitts (1943). These pre-eminent scientists were known to be well acquainted with one another, jointly attending meetings and sharing in discussions (Gefter, 2015). However, perhaps most telling of their extensive mutual influence is simply the language of their work – von Neumann's EDVAC report was written using the same logical form and terminology as McCulloch and Pitts's article, describing the components of the EDVAC as 'organs' and 'neurons'.

However, John von Neumann was not alone in his natural inspiration among the early computer pioneers. Indeed, Alan Turing, considered by many to be the father of theoretical computer science and artificial intelligence, based much of his early work on his knowledge of the organization of the human brain. In his work 'Intelligent Machines' (Turing, 1948), he states '[t]he analogy with the human brain is used as a guiding principle' in referring to his proposed 'unorganized machines', a work which is now acknowledged to be the first description of a trainable artificial neural network. Perhaps even more famously, in introducing the model which now forms the foundation upon which all modern artificial neural networks are built, Frank Rosenblatt's 'Perceptron', Rosenblatt explicitly proposes "[t]he theory serves as a bridge between biophysics and psychology", and that "[the] perceptron is first and foremost a brain model, not an invention for pattern recognition" (Rosenblatt, 1958).

As computational power increased over the intervening decades, this natural inspiration similarly shifted from theory into practice. The computational primitives introduced by these early pioneers were shown to be highly flexible, powerful, and degraded in performance gracefully, eventually allowing them to overtake the less neurobiologically motivated 'symbolic' models which temporarily had risen in popularity. As these ideas of parallel distributed processing began to become more widely accepted, work to make such systems even more brain-like continued as a means to address the limitations of existing systems. As a primary example of this, in 1980, in order to address the fragility of existing artificial neural networks with respect to small image shifts and deformations, Kunihiko Fukushima introduced a new model under the name the 'Neocognitron' (Fukushima, 1980). The model was built to have localized and repeated feature detectors across space to mimic forms

of observed organization in the natural visual systems of many species' brains. It further integrated iterative alternating layers of computation and pooling, directly inspired by the simple and complex cells of David Hubel and Torsten Wiesel discovered less than two decades earlier (D. H. Hubel and T. N. Wiesel, 1962). This architecture was the first of what eventually became known as the class of convolutional neural networks, arguably one of the most successful artificial neural network architectures to date, and resulted in the explosive growth of deep learning in the 2010's.

From this historical overview it is clear that many of the people who once reshaped the world with revolutionary computing paradigms did so by observing and abstracting the elegant solutions nature had painstakingly developed over the millennia before them. In this thesis, we will argue that we have not reached the end of this path, and that by continuing to follow the example set by these visionaries, we are likely to discover not only how to build truly intelligent machines, but also the profound truth underlying all forms of intelligence.

1.2. The Growing Divide of Efficiency and Generalization

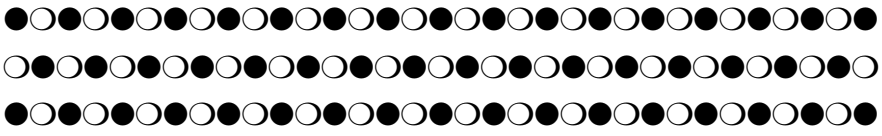
In recent years, by building on the shoulders of these naturally inspired pioneers, artificial intelligence has continued to flourish at an astonishing rate. In 2012, the work of Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton ushered in the so called 'ImageNet Moment', where artificial neural networks were shown to be significantly more capable at object recognition than all systems designed by computer vision experts of the time (Krizhevsky, Sutskever, et al., 2012). Similarly, in 2016, AlphaGo beat the worlds best players at one of the most computationally complex board games created by human kind, Go (Silver et al., 2016). By the same time, generative adversarial networks demonstrated the ability to generate realistic human faces (Radford, Metz, et al., 2016), and these were subsequently followed by text-to-image generative models such as DALL-E in 2021 generating novel images directly from text descriptions, such as 'a teddy-bear on the moon' (Ramesh, Pavlov, et al., 2021). Finally, to the surprise of many including machine learning experts themselves, the development of 'large language models' appeared to pass the Turing test convincingly in 2022 (Thoppilan et al., 2022) with chat-bots becoming a modern commodity shortly thereafter.

As the successes of artificial intelligence have compounded, driven primarily by monumental engineering efforts devoid of explicit natural inspiration, one is unsur-

prisingly driven to wonder: is there still anything to be gained by studying natural intelligence? In this thesis I will argue that the answer is unequivocally yes, and that there are two fundamental differences between natural intelligence and its artificial predecessor which I believe are unlikely to be satisfactorily resolved by following current machine learning trajectories. These differences are namely: data efficiency, and generalization performance.

Efficiency

Let's consider the groundbreaking Go-playing program, AlphaGo. Its human opponent in the highly publicized match, Lee Sedol, is estimated to have played on the order of 100,000 games throughout his training lifetime, ultimately achieving the highest possible rank of 9th dan. Comparatively, Alpha Go is believed to have learned from nearly 100 million or more games altogether (Lake et al., 2017). To understand the scale of this difference. If 1,000 games were represented by a single character (❍ or ●), a human player would require the following amount of training:



For AlphaGo, the number of characters would fill more than 50 pages. For the reader's convenience we have omitted printing this explicitly, however the reader is encouraged to turn to page 55 for a visualization of the relative thickness of this much paper. In a more controlled setting, such discrepancies between the speeds of human and artificial learning efficiency have been studied experimentally by scientists such as Lake et al. (2017). In their work, these authors report that humans are able to learn to effectively play a suite of Atari games in just two hours, reaching a level of performance that took modern deep reinforcement learning agents an equivalent of 924 hours of game time to learn (Schaul et al., 2016).

However, it is well known that reinforcement learning agents are not the only algorithms which require orders of magnitude more data than humans to perform comparably. Large language models (LLMs) are trained on nearly the entirety of the internet in order to be able to answer natural questions reasonably. LaMBDA, an early LLM from Google, is stated to have been trained on 1.56 trillion tokens extracted from public dialog and text (Thoppilan et al., 2022). As a point of

comparison, it is estimated the average English non-fiction reading speed is roughly 240 words per minute (Brysbaert, 2019), meaning that it would take the average human 18,550 years of reading non-stop, 16 hours per day, to ingest the same amount of data. Even more recent models such as GPT-4 are rumored to have been trained on near 13 trillion tokens, equating to more than 150,000 years of human reading (Schreiner, 2023). Although these systems arguably also possess significantly more information internally than any living human being, it is clear that they do not behave in nearly the same manner as a wise sage who has managed to elude death long enough to read an equivalent amount of text. In the following subsection we will argue that this kind of behavioral mismatch is a telltale sign of at the second greatest distinction between natural and artificial intelligence, a difference in how these systems learn from their necessarily finite training set (albeit large but still finite) and ultimately generalize to new situations.

Generalization

It is clear from even cursory uses of modern language models or image generative models that these models do not generalize in any manner that resembles how humans generalize. For example, although modern text-to-image generation programs are able to generate highly photo-realistic images which appear to largely match their text-based prompts, their ability to generate slightly unorthodox images is still surprisingly lacking. Consider the examples shown in Figure 1.1 from the state of the art generative model known as DALL-E 2 (Ramesh, Dhariwal, et al., 2022). Although the images are highly photo-realistic, contain the mentioned objects, and are far better than anything most humans could draw or even create digitally, they appear to have fundamentally misunderstood core elements of the text prompt, such as relatively simple relations between objects. Consider a small child as a point of comparison. Although it is clear that a child would never be able to draw the detailed lighting, shading, and texture of these images at a similar level of complexity, a child would likely be able to at least correctly draw a stick-figure banana holding a stick figure monkey without getting permanently confused. This counter-intuitive performance reminds us of Moravec's paradox (Moravec, 1990) – when it comes to artificial intelligence (AI) research: *"the hard problems are easy and the easy problems are hard"* (Pinker, 2007). We've developed machines that appear to surpass Leonardo Davinci's level of control over lighting and perspective yet simultaneously fail at stacking blocks in the same manner as a toddler.

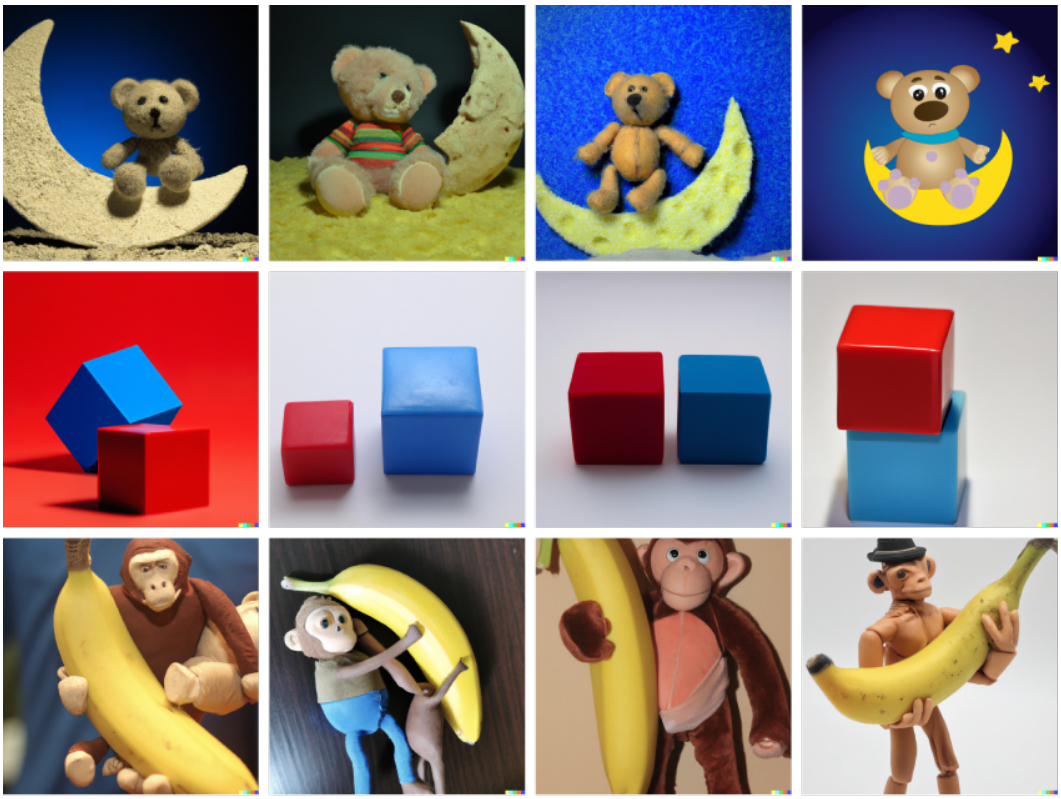


Figure 1.1: Samples from text-to-image generation program DALLE-2. The prompts are: (top) "a teddy bear on the moon", (middle) "blue cube on top of a red cube", & (bottom) "a banana holding a monkey". We see that while the model can generate incredibly realistic images of relatively novel objects, it often fails to understand basic relations between objects.

What is it about these systems that makes their generalization so counter-intuitive to us? What makes some tasks so easy for us, and so challenging to reproduce in silicon? Although with the current rate of development of artificial intelligence, future readers of this work may find that engineering efforts have largely resolved the differences listed above, there is a common underlying theme from learning theory which unites these differences and similarly relates them to our original question of why we should still be interested in natural intelligence – that underlying theme is inductive bias.

1.3. Inductive Bias: The Great Arbiter of Learning

The ability to generalize from a limited set of examples to underlying principles is a defining characteristic of what we commonly refer to as ‘learning’. For instance, a child is said to have learned the concept of ’dog’ when they associate every canine

they encounter with the term ‘dog’ and not just the familiar pet at home. Similarly, they have learned addition when they can compute the sum of any two numbers, not just the numbers on their homework. This process of generalization is often called inductive reasoning, and performing this generalization well is one of the ultimate challenges of those aiming to design artificial learning systems.

At a theoretical level, in order for any learning system to generalize beyond the examples it has been trained on, some assumptions must be made which provide a basis upon which the learner can begin its inductive generalization. In fact, when no assumptions are made, it is known to be theoretically impossible to generalize beyond one’s training set (Wolpert, 1996).

To shine a light on these hidden assumptions that we almost always take for granted, we will borrow an example from the PhD thesis of Joshua Tenenbaum (1999). The example can be loosely paraphrased as follows: Imagine I have created a computer program which generates a set of hidden numbers from a simple rule, for example ‘all multiples of 3’ or ‘all odd integers less than 14’. In all cases we restrict the numbers to be integers and less than 100. The program then presents you with a random subset of the total set of numbers generated by this rule, and your job is to guess which other numbers are most likely to also be in the set, i.e. the remaining members generated by the rule. Consider then, being presented with the number 16. From this number alone, one may guess that the numbers 17 or 18 are more more likely companions to 16 than the number 97. However, what is it that is causing this ‘gut feeling’? What is it that is allowing us to even attempt to infer a general underlying rule from this single example? Do we have some weak assumption that 16 is somehow more related to 18 than 97? What would happen if we had no such assumptions?

Consider the even stronger example from the same task paradigm: after being presented with 16, you are provided the additional full set of numbers: 16, 8, 2, and 64. In the work of Tenenbaum (1999), it was shown that the vast majority of respondents then asserted the remaining most likely set elements to be 4 and 32. Intuitively, this conclusion seems highly reasonable, and I would believe that most readers of this work will also likely agree with it. However, when examined a bit more closely, a crack appears in the facade. There are practically an infinite number of hypotheses which fit this data perfectly, why should we assume powers of two? For example, "all integer multiples of 2, except for 6", or "all integers except for 1, 3, 5, 6, 7", or even more simply, "all integers less than 100". What then makes

"all powers of 2" the most compelling conclusion? Tenenbaum argues convincingly through theory and case studies that this is due to human concept learning operating within a Bayesian framework; i.e. the powers of two are indeed the most likely, assuming a prior distribution over programs and a random sampling procedure for producing the numbers.

The point of this example for our purposes however is not to emphasize the potential Bayesian nature of natural intelligence, but rather to get the reader to imagine the setting where we had no compulsions towards any conclusion, if we had no 'gut feeling' towards which numbers felt likely to belong within the set. If we believed the number 97 was equally as likely as 32 to join the aforementioned set, how then could we make any predictions about any of the digits outside the given set? It seems our ability to generalize from this limited data to a general rule would be entirely lost. This example demonstrates the core significance of inductive bias in learning: without any inductive bias, there is really nothing that can be said about new data points that have not been seen before, and thus any reasonable predictions outside the training set become impossible.

Beyond simply making generalization possible however, inductive biases are known to actually maintain a grip on how a learning system generalizes throughout the training process, and how much data is therefore required to do so. For example, consider that you were predisposed to only consider two possible hypotheses: 'all even numbers', or 'all odd numbers'. In such a setting, you would only require a single example in order to form your final conclusion about what you believe the underlying rule to be – if you see a single even number, you can confidently guess all numbers are even, and vice versa. However, in this case the double-edged sword of inductive bias is strikingly apparent; unless the computer program also was restricted to such a limited set of hypotheses, our guess would likely be grossly incorrect. This example demonstrates the simultaneous power and perils of inductive biases: they can dramatically facilitate both generalization performance and data efficiency if they are properly suited to the target domain, but if they are mis-specified they can make correct generalizations impossible.

Considering that the two primary distinctions between natural and artificial intelligence we outlined in Section 1.2 are heavily related to generalization and data efficiency, it makes sense to begin our search for the source of such distinctions by a search for inductive biases. But how do we choose the appropriate inductive biases? As demonstrated by the final example above, the answer turns out to be

relatively simple, and brings us back to our opening question about why we should study natural intelligence: the optimal inductive biases of a learning system are those harmoniously tuned to their environment. Mother Nature, renowned for her fine-tuning abilities, surpasses any mechanic we know. So, in this regard, if we listen closely to her wisdom, what insights might she reveal?

In the next chapter we will describe with historical and mathematical support why we believe some of these insights may lie in the observed structure of natural neural representations. Through contrasting examples of structure in natural and artificial systems we will attempt to motivate each of the three parts of our thesis, and follow with an overview of the research questions this thesis ultimately aims to address.

Chapter 2

MOTIVATION

Flickers of candle light wash over the multicolored brick walls of an old English town-home. A young physician, Richard Caton, leans forward in his chair holding one electrode in each hand, staring in anticipation at the exposed brain of an immobilized rabbit which lies on the desk before him. The electrodes are attached to a galvanometer, developed only a decade earlier by Lord Kelvin to amplify weak electrical currents and make them visible through the movement of a spot of light projected on a distant surface. In a moment of silence, Caton touches the electrodes to the surface of the brain – the spot moves, a current is registered. Caton then moves the flame from across his desk into the sight of the rabbit – the spot moves again, this time even more dramatically. In this darkened room in the English countryside, Caton has just measured the how a brain uses electricity to represent visual experience, perhaps for the first time ever.

These results were published August 28th, 1875 in the British Medical Journal, and represent what can now be recognized as the first known recording of *neural representations* of a visual stimulus (Caton, 1875). Specifically, in a mere paragraph encompassing only a quarter-page, the article reports "In every brain hitherto examined, the galvanometer has indicated the existence of electrical currents... Impressions through the senses were found to influence the currents of certain areas." With time, our ability to measure and quantify the precise form of these representations has increased dramatically. In doing so, our understanding of how the activity of neurons corresponds to different stimuli and internal states has similarly progressed.

One of the greatest insights garnered from this increased measurement ability is the understanding that natural neural systems have surprisingly structured activity both in space and in time. Even before Canton's discoveries, it was observed that different localized areas of the brains of various animals appeared to correspond to specific functions. For example, Ferrier (1874) published early results demonstrating that electrical stimulation of specific regions of the brains of cats, rabbits and dogs could be shown to be more likely to induce involuntary movement than the stimulation of other areas. Later, scientists such as Hubel and Weisel discovered structure not

just in large scale patterns, but at the level of individual neurons (D. H. Hubel and T. N. Wiesel, 1959), and perhaps most importantly, in between layers of neurons (D. H. Hubel and T. N. Wiesel, 1962). These findings sketched out the idea that the organization and structure of neural connectivity could be an essential part of how the brain represents and learns from visual stimuli, and the scientists ultimately received the Nobel Prize for their work in 1981. Since then, the study of the organization of the brain has come to encompass entire fields and conferences[1], and only continues to grow more advanced with functional magnetic resonance imaging (fMRI) and multi-electrode recording. This structure has attracted the intrigue of generations of scientists, with many wondering what computational role these systems may perform.

In the history of machine learning, this natural neural structure has been often imitated, frequently contributing to some of the greatest advances in artificial intelligence to date, such as the multilayer perceptron of Rosenblatt and the primordial convolutional neural network of Fukushima. In this thesis, we question what inductive biases may be waiting to be discovered in the remaining structural differences between natural and artificial neural network representations. To begin this journey, we will first examine well known forms of representational structure in both natural and artificial systems aiming to inspire the reader to draw the same intuitive connections which have motivated this research.

2.1. 'A small machine that looks after all values of a given variable'

As alluded to above, one of the most striking findings from neuroscience is the apparent spatial organization of neurons according to function or responsiveness, a property known as topographic organization. This organization has been measured across a diversity of different brain regions and with respect to an even broader range of stimuli. A founding example of this type of organization is what is known as retinotopic organization, whereby the visual field is mapped to the surface of the cortex in a spatially continuous manner.[2] Similarly, D. H. Hubel and T. N. Wiesel (1962) discovered that the visual cortex of many species has topographic organization not just with respect to visual location, but also with respect to visual features such as orientation. Precisely, they found that neurons which were located closer to

[1] See the Organization for Human Brain Mapping conference.

[2] This can equivalently be seen as the co-location of neurons with similar spatial receptive fields, where the receptive field of a neuron is defined as the area of the visual field in which it is most responsive to stimuli.

one other across the surface of the cortex were more likely to respond strongly to similar orientations of lines. Extrapolating this simple principle across the entirety of the primary visual cortex, one finds a patterned structure of smoothly varying orientation selectivity, eventually covering all orientations, and sometimes resulting in 'topological defects' whereby selectivity for all orientations come together at a single point, called a 'pinwheel' (Koulakov and Chklovskii, 2001). Hubel and Wiesel also noted that it is possible to subdivide the cortical surface into groups (which they called 'hypercolumns') such that each group contains roughly a full set rotation angles. They suggested that these groups may form a fundamental building block of the cortex, calling each 'a small machine that looks after all values of a given variable' (David H Hubel and Torsten N Wiesel, 1974a).

In the context of a machine learning system, such a group of neurons with organized selectivity would likely have beneficial representational properties. For example, if the input stimuli were to rotate at a specific location, the activity in the network could similarly be observed to 'rotate' *within* one of these groups. Then, if the output of the network needed to be invariant to the orientation of the input, meaning it only needed to know the presence of the input regardless of orientation, the network could simply take the sum over the entire hypercolumn/group.[3] In fact, these types of structured models have already been studied in the machine learning literature under the names of Capsule Networks (Geoffrey E. Hinton, Krizhevsky, et al., 2011b) and Equivariant Neural Networks (T. Cohen and Max Welling, 2016a). In such models, the network connectivity is specifically structured such that when a desired transformation is applied to the input, there is a known predictable transformation which occurs in the corresponding output of the network. Furthermore, in many versions of such models, this output transformation has exactly the same core property of that of Hubel and Wiesel's hypercolumns: neural activity only shifts within a 'capsule', but not between 'capsules'. In other words, each capsule or equivariant group of neurons is exactly 'a small machine that looks after all values of a given variable'.

Such a connection naturally leads us to wonder: Could the orientation hypercolumns observed by Hubel and Wiesel be instantiations of equivariant capsules? Furthermore, could the abstract mechanism which is used to induce such hypercolumn structure in natural systems be equally beneficial for inducing equivariant structure in artificial neural networks?

[3]This is a well known technique to form an invariant representation from a set since the sum does not change no matter how the elements are permuted or in this case 'rotated'.

2.2. La Nouvelle Vague

In addition to topographic organization, one of the earliest observed properties of electrical signals in the brain was their oscillatory behavior. In one of Richard Caton's first articles he writes "The current is usually in constant fluctuation; the oscillation of the index generally small, about twenty to fifty degrees of the scale. At other times, great fluctuations are observed, which in some instances coincide with some muscular movements or change in the animal's mental condition" (Caton, 1877). With time, Hans Berger, the inventor of the electroencephalogram (EEG), measured and characterized these 'fluctuations' as alpha and beta oscillations, now known to be formed from the coordinated activity of a multitude of neurons firing in synchronous structured patterns (Berger, 1929). As measurement capabilities have contintued to advancę, scientists have begun to discover that many of these oscillations are actually best described by traveling waves of activity across the surface of the cortex (Muller, Chavane, et al., 2018).

Recently, in the machine learning literature, T. Konstantin Rusch and Mishra (2021a) introduced the Coupled Oscillatory Recurrent Neural Network (coRNN) directly inspired by the observed oscillatory structure in natural activity. The authors demonstrated that by parameterizing a recurrent neural network as a system of coupled harmonic oscillators, the dynamics of these networks were provably more stable over time, thereby yielding significant performance improvements on very long sequence modeling tasks. Such long tasks are well known to be the Achilles' heel of standard recurrent neural networks due to the well known 'exploding and vanishing gradient problem'. This model gives an inspiring example for the potential of natural inspiration in artificial systems; however, there still remains a lack of models which exhibit the wave-like dynamics now known to be closely linked to this oscillatory activity. Might there be additional sequence modeling benefits to extending this oscillatory structure over the spatial dimensions as well?

As described in Section 2.1, there is increasingly large set of 'equivariant' deep neural networks in the machine learning literature which leverage highly structured connectivity patterns and weight sharing schemes in order to control how the output of these networks changes with respect to input transformations. In many of these models, the output can be seen to smoothly 'flow' between the neurons within an 'equivariant capsule' when the desired transformations are applied to the input.[4]

[4]The reason for this 'smooth flow' is that these equivariant networks are built to explicitly preserve the abstract properties of the transformations groups of interest, including the topology of

Could traveling waves then serve as a more flexible and organic mechanism to encourage such a smooth flow of activity in the latent space of neural networks? In other words, if we were to build neural network models with traveling wave dynamics, would we then observe structured representations similar to equivariant neural networks?

2.3. Structured Organization and Connectivity

As described above, it is becoming increasingly clear that the cortex is highly structured over both space and time. This structure manifests itself in the form of functionally localized selectivity, as well as through more dynamic patterns such as traveling waves. While so far in the thesis we have alluded to how mature 'fully learned' structure in natural systems may relate to structure in the machine learning literature, this discussion has bypassed a key point: the process of learning these representations in the first place. We are therefore naturally inclined to wonder, is there a more fundamental relationship between this structure and the process of learning?

In machine learning, there is a learning framework known as self-supervised learning which is commonly said to learn directly 'from the structure in the data itself' (LeCun and Misra, 2021). Explicitly, this is typically accomplished by performing transformations to the input, and then using the discrepancies between the original and transformed inputs to determine the essential features which span the high level concept space. Mathematically, functions which are able to preserve structure from an input space to an output space are known as homomorphisms, and interestingly, the equivariant networks we have mentioned in the previous sections are known to fall into this class of functions. Considering self-supervised learning is accomplished by leveraging transformation structure in the data, if our neural network was known to be one of these homomorphisms, could the network instead perform self-supervised learning by simply leveraging the structure within its own output space? Could this be an alternative local learning mechanism for artificial neural networks which could avoid both the need for supervsision and end-to-end backpropagaion of error signals?

---

these groups. In doing so, transformations which are neighbors in this topological space will be mirrored by neural activations which are similarly 'neighbors'. Of course, this asserts that neurons are spatially grouped together by these topological neighborhood.

2.4. Contributions

Consolidating the above connections we have drawn between the structure of natural and artificial neural network representations, we can begin to formulate the primary research questions this thesis aims to address. To begin, we first organize this work into three parts motivated by the three sections above: (I) a study of spatial structure in neural representations, (II) an extension to spatio-temporal structure, and finally (III) a study of how learning may be performed through the use of such structured representations.

In part one of our work, we will mainly aim to answer the research question:

Research Question 1: What role does topographic organization play in the computational functions of the brain?

To accomplish this, we will introduce a new model which yields topographic (spatial) structure in neural representations through the framework of probabilistic generative modeling. We will further show how, through training, such models learn to exhibit topographic organization reminiscent of the higher visual cortices of mammals. Combined, this work provides a hypothetical mechanism by which topographic organization could be achieved in a deep generative modeling framework, as well as a computational argument for the adaptive benefits of such structure from an information theoretic perspective.

In part two, we will aim to generalize the structure we study from purely spatial structure to spatio-temporal structure. Similar to part one, our initial research question will be quite broad:

Research Question 2: Does spatio-temporal structure play a role in the computational functions of the brain?

In addition to this question, we will aim to further answer the reciprocal question:

Research Question 3: Can natural spatio-temporal structure be efficiently and beneficially implemented in deep artificial neural network architectures?

To answer these questions, we will again introduce a suite of models aimed at reproducing aspects of spatio-temporal structure observed in natural systems. First, we will show how this type of structure has strong ties with topographic organization, thereby effectively entangling the answers to these three questions. We will then demonstrate empirically how spatio-temporal structure in the brain directly relates to equivariant structure as introduced in the machine learning community. By do-

ing so, we show that spatio-temporally structured models are better able to model datasets with strong symmetries when compared with similar models which lack such organization. Finally, we will demonstrate for the first time how such structure can be used to efficiently and robustly encode memories in recurrent neural network architectures, thereby providing computational evidence for theories from neuroscience.

In the final part of this work, we will consider the value of this structure from an alternative perspective. Specifically, we will ask:

Research Question 4: Can spatio-temporal representational structure be leveraged to perform efficient and local learning without labeled data?

We will address this topic by introducing a new self-supervised learning framework, which we call Homomorphic Self-Supervised Learning, which serves to unify existing self-supervised learning objectives through the lens of structured representations.

In the remainder of this thesis we will investigate these questions and return to them specifically in the conclusion to discuss our findings. Primarily, the contributions of this work are therefore two-fold and mirror the reciprocal symbiosis of the studies of natural and artificial intelligence. First, this work strives to improve artificial neural networks through the development of novel architectures which incorporate natural structure. Second, in doing so, this work strives to yield an improved understanding of the computational roles of this structure in natural systems.

Although in this work we assume a basic familiarity with calculus, linear algebra, probability theory, and some machine learning terminology, we still aim to make the findings accessible to a wide range of backgrounds given the interdisciplinary nature of the work. Before we delve into our own contributions and thereby attempt shed some light on these questions, we will first provide a brief review of artificial neural networks that we will use in this study, and how their core components can be seen to relate to common biological abstractions.

Chapter 3

BACKGROUND

In this work, we will focus on an abstracted picture of the brain that is favored by some theoretical and computational neuroscientists. This abstraction is primarily focused on modeling the neocortex of mammals using the deep artificial neural networks built off of early brain models such as the perceptron. In doing so, we will abstract away the layers of the cortex which penetrate down towards the center of the skull and instead consider it instead as a 2-dimensional sheet. We will furthermore abstract away many specifics of cellular neurobiology, including the distinction of exictatory and inhibitory cells, as well as their continuous time 'spiking' behavior, instead interpreting the responses of our artificial neurons as a population rate-based code.[1] By constructing such an abstract system, we will then have the computational means necessary to study the task-relevant properties of natural neural representational structure at scale.

3.1. Artificial Neural Networks

The types of artificial neural networks we will employ throughout this work are formed from iterative alternation of linear transformations and non-linear 'activation functions'. At the simplest level, the standard artificial neural network layer $f_l(\boldsymbol{x})$ may be written as $\sigma(\boldsymbol{W}_l\boldsymbol{x}_l)$, with input vector $\boldsymbol{x}_l \in \mathbb{R}^n$, activation function σ, and connectivity matrix $\boldsymbol{W}_l \in \mathbb{R}^{m\times n}$ (assumed to encompass the usual bias term $\boldsymbol{b}_l \in \mathbb{R}^m$). Such a layer is said to have m neurons (corresponding to the m output values), and $m \times n$ weights (abstract synapses). Through composition of multiple of such functions, we arrive at one of the primordial neural network architectures, the multilayer perceptron (Rosenblatt, 1958):

$$\mathrm{MLP}(\boldsymbol{x}) = f_L \circ f_{L-1} \circ \ldots \circ f_0(\boldsymbol{x}_0) \tag{3.1}$$

The input-output mapping specified by the function $\mathrm{MLP} : \mathcal{X} \to \mathcal{Z}$ is then typically used to approximate some desired mapping between an input space $\mathcal{X}$, such as the

[1] With rate-based coding, we loosely interpret the real valued activation of a neuron to refer to the frequency of spikes per time interval. For artificial neural networks with negative-valued activations, this interpretation becomes even more loose.

space of 32×32 pixel images, and the desired output space $\mathcal{Z}$, such as the space of object categories assigned to each image.

In such networks, choices of activation functions and connectivity constraints can determine valuable inductive biases of the network. A common activation function in the majority of recent work is the Rectified Linear Unit activation (ReLU: $\max(0, x)$, (Nair and Geoffrey E. Hinton, 2010)) due to its observed beneficial optimization properties in the context of deep neural networks (i.e. its gradients do not vanish as quickly with depth when compared with other saturating non-linearities such as the hyperbolic tangent). When we refer to the parameters of the model, we refer to the weights $\boldsymbol{W}$, which are usually randomly initialized from a normal distribution with small variance and subsequently learned through gradient-based training. One additional type of common neural network layer is known as a 'pooling layer' which can be seen as a form of dimensionality reduction by typically taking the average or maximum of the activations in a given layer or region and passing those on to the next layer. To keep our formulation general, we note that such layers can additionally be included in Equation 3.1 above by simply setting the matrix $\boldsymbol{W}$ above to a fixed linear operation such as averaging, or the identity, and then setting the activation function σ to the appropriate non-linear counterpart, such as the identity or the max operation to construct average and max pooling layers respectively.

Recurrent Neural Networks

The framework described above can considered to be independent of time. It is what is known as a feed-forward neural network, in that all activity propagates from the input through to the output layer without any loops known as recurrent connections. If we add such connections, we arrive at one of the second fundamental neural network architectures that we will work with in this thesis, the recurrent neural network. Specifically, such networks can best be described by a time-varying hidden state which we denote $\boldsymbol{h}_t \in \mathbb{R}^m$, and a sequence of inputs which we denote $\boldsymbol{x}_t$ for $t = 0$ to T. The hidden state is typically initialized to the zero vector for the first time step ($\boldsymbol{h}_0 = \boldsymbol{0}$) and subsequent timesteps are then recursively computed as:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t) \tag{3.2}$$

where the hidden state is now connected to itself in time through the recurrent connectivity matrix $\boldsymbol{U} \in \mathbb{R}^{m \times m}$. Again, the activation function σ is a design choice

which is typically chosen to be a bounded sigmoidal function such as a hyperbolic tangent function or logistic function, or the ReLU function for the same beneficial gradient propagation properties (Q. V. Le et al., 2015).

As we will see later in Chapters 6 & 7, one useful way to think of recurrent neural networks is as time-discretized versions of differential equations. For example, Equation 7.1 above can be equivalently expressed as the following first order ordinary differential equation (ODE):

$$\frac{\partial \boldsymbol{h}(t)}{\partial t} = -\gamma \boldsymbol{h}(t) + \sigma(\boldsymbol{U}\boldsymbol{h}(t) + \boldsymbol{W}\boldsymbol{x}), \qquad \boldsymbol{h}(0) = \boldsymbol{0}. \tag{3.3}$$

Through the Euler Method we can then numerically integrate this ODE over discrete timesteps to arrive at:

$$\boldsymbol{h}_t = (1 - \gamma)\boldsymbol{h}_{t-1} + \Delta_t \sigma(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t). \tag{3.4}$$

This formulation allows us to make a more direct connection between how we describe natural neural networks using known laws of physics and how we may implement them efficiently in an artificial setting. By making this connection explicit, we are given additional mathematical control over the integration scheme, the parameters such as γ and Δ_t, and further able to easily develop novel recurrent neural network architectures inspired directly by common ODEs. In Chapters 6 & 7 we will make use of this fact explicitly to implement recurrent neural networks which act like simple wave equations ($\frac{\partial \boldsymbol{h}}{\partial t} = c\frac{\partial \boldsymbol{h}}{\partial x}$) and harmonic oscillators ($\frac{\partial \boldsymbol{h}^2}{\partial t^2} = {}^{-}\frac{k}{m}\boldsymbol{h}$).

Convolutional Neural networks

As mentioned in the introduction, another common architectural choice for artificial neural networks when working with structured input data is to employ what are known as convolutional neural networks (CNNs). Such networks often have an analogous structure to that of Equation 3.1, with the core difference being in how the connections $\boldsymbol{W}$ are formed. Specifically, in the case of the MLP, the matrices $\boldsymbol{W}$ are dense, meaning that every neuron from layer $l - 1$ is connected to each neuron in layer l with its own independant weight value (parameter). Such layers are known as 'fully connected'. In contrast, convolutional neural network layers are not fully and independently parameterized, but instead are formed from a single set of weights which are shared between all neurons in a given layer. Explicitly, these shared weights are called the 'kernel' of the convolutional layer, and typically consist of a number of input and output channels which allow for specification of the

number of neurons in each layer. For simplicity, let us consider a one-dimensional kernel of size 3 with only a single input and output channel: $\boldsymbol{w} \in \mathbb{R}^3 = [w_0, w_1, w_2]$. Additionally, to avoid boundary effects, let us define the convolution to be circular, meaning that the convolution wraps around the edges of the input vector. The operation of such a convolutional neural network layer can then be written as:

$$\boldsymbol{w} \star \boldsymbol{x} = \begin{bmatrix} w_0 & w_1 & w_2 & 0 & \cdots & 0 \\ 0 & w_0 & w_1 & w_2 & \cdots & 0 \\ 0 & 0 & w_0 & w_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ w_2 & 0 & 0 & 0 & \cdots & w_1 \\ w_1 & w_2 & 0 & 0 & \cdots & w_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} w_0 x_0 + w_1 x_1 + w_2 x_2 \\ w_0 x_1 + w_1 x_2 + w_2 x_3 \\ w_0 x_2 + w_1 x_3 + w_2 x_4 \\ \vdots \\ w_0 x_{n-2} + w_1 x_{n-1} + w_2 x_0 \\ w_0 x_{n-1} + w_1 x_0 + w_2 x_1 \end{bmatrix} \tag{3.5}$$

Due to the sharing of the kernel weights $\boldsymbol{w}$ over all dimensions of the input $\boldsymbol{x}$, this operation can be interpreted as 'sliding' the kernel over the input and computing repeated inner products for each successive shift. More commonly, as in the signal processing literature, this operation is known as a cross-correlation and is defined for a single element i of the output as follows:

$$[\boldsymbol{w} \star \boldsymbol{x}]_i = \sum_{j=0}^{n-1} x_j w_{(j-i)\%n} \tag{3.6}$$

where the filter $\boldsymbol{w}$ is assumed to be equal to the size of the input $\boldsymbol{x}$ (or otherwise zero-padded), and the subscript $(j - i)\%n$ denotes modulo n, indicating that the convolution is circular. In this work we will use common deep learning terminology and refer to this operation as a convolutional layer.

Such layers have known connections with the retinotopic organization described above. Specifically, if we consider the dimension(s) over which the convolution operates to be equivalent to spatial dimensions along the cortex, then the above operation can be seen as equivalent to the combined ideas that: (I) each neuron has a localized receptive field in the input (i.e. each neuron only processes a limited subset of adjacent x_i), (II) neurons with similar receptive fields are located next to one another in the brain, and (III) the same selectivity patterns (feature detectors) are repeated throughout the spatial extent of the input. When combined with the common local-pooling operators described above, this architecture begins to resemble circuits first studied by Hubel and Wiesel in the early visual cortex, and this is precisely the structure that Fukushima aimed to emulate with his early Neocognitron architecture.

3.2. Equivariant Neural Networks

As noted earlier in this work, the convolutional neural network has demonstrated itself be one of the most successful inductive biases in the machine learning toolkit to date. With time, the fundamental theoretical principles behind this success have been unraveled, and one which has continued to stand out is the concept of translation equivariance. In a general sense, equivariance can be viewed as an inductive bias towards representations with geometric group structure (i.e. symmetries). Slightly more precisely, equivariance of a function can be understood to mean that for a given set of input transformations of interest, there is a corresponding known and well-behaved transformation of the function's output in the output space. In the language of group theory and representation theory, this is commonly written as $f(\tau_g[\boldsymbol{x}]) = \Gamma_g[f(\boldsymbol{x})]$ for a function f, a transformation g, and the associated input and output representations of that transformation τ_g and Γ_g respectively. In the case of convolution, the operation of translating (i.e. shifting) the input can be seen as equivalent to translating the output of the network in feature space.

Another intuitive way to think of equivariance, which will come up multiple times in this thesis, is that it implies the function (or neural network) can be seen to commute with the transformation operator. In other words, it does not matter if one first transforms the input and then passes it through the function ($f(\tau_g \boldsymbol{x})$), or if one instead passes the un-transformed input through the function and then applies the transformation to the output ($\Gamma_g[f(\boldsymbol{x})]$), the result will be the same. We can draw this in the form of a diagram as follows:

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{\tau_g} & \mathcal{X}' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\ \mathcal{Z} & \xrightarrow{\Gamma_g} & \mathcal{Z}' \end{array}$$

We see that, starting from $\mathcal{X}$, no matter whether we follow the upper path or the lower path, we arrive at the same point $\mathcal{Z}'$. As we will see throughout this work, many of our figures will resemble this type of commutative diagram to show that our network does indeed commute with the observed transformations.

In the years since the convolutional layer's widespread adoption, significant work has gone on to generalize the set of transformations to which networks can be made equivariant. These supported transformation groups now include rotations and mirroring (T. Cohen and Max Welling, 2016b), scaling (D. Worrall and Max Welling, 2019a), and ultimately any continuous compact Lie group (Finzi, Stanton,

et al., 2020). These networks have been demonstrated both empirically (T. Cohen and Max Welling, 2016a; D. E. Worrall et al., 2017; Veeling et al., 2018; Pol et al., 2020) and theoretically (Elesedy and Zaidi, 2021; Farrell et al., 2021; Bordelon and Pehlevan, 2022) to improve data efficiency and generalization performance when the transformation groups they incorporate are reflected in the data they are aiming to model.

Example: Translation Equivariance

To briefly sketch how the generalization of equivariance from translation to larger groups is accomplished, let us consider the elements of the 2-D translation group more abstractly as a pair of integers $g_{i,j} = (i, j) \in \mathbb{Z}^2$ denoting the x and y displacement of a given translation operation. To form a group, we need to combine this set of elements with a group operation (denoted $\cdot$) which takes two group elements and returns another element of the group: $g_{i_1,j_1} \cdot g_{i_2,j_2} = g_{i_3,j_3}$. This property is known as closure and can be seen as the first group axiom: (i) when combining any two group elements, the output is always another element in the group. Furthermore, the combination of the elements and the operation must satisfy the remaining group axioms ((ii) associativity and the existence of (iii) identity & (iv) inverse elements). In the case of translation, if we define the group operation as element-wise addition (+), we can see that indeed this combination $\mathcal{G} = (\mathbb{Z}^2, +)$ satisfies the conditions for a group, i.e. (i) $i + j \in \mathbb{Z} \; \forall \; i, j \in \mathbb{Z}$ (ii) $(i + j) + k = i + (j + k)$, (iii) $i + 0 = i$, & (iv) $i + -i = 0$.

How then does this abstract definition assist us in generalizing convolution? First, consider the definition for the standard convolution of a filter $\boldsymbol{w}$ (again with a single channel) with a 2-dimensional signal $x(i, j)$. In this example we will use the common procedure of indexing the signal with parentheses to denote that it is actually a function defined over the space $\mathbb{Z}^2$. We will leave the filter indexing with subscript notation for continuity with the above equations, however a similar functional definition of $\boldsymbol{w}$ is equally valid and often commonly seen in the equivariance literature. With this in consideration, we can write the 2-dimensional convolution as:

$$[\boldsymbol{w} \star \boldsymbol{x}] (i, j) = \sum_{(k,l) \in \mathbb{Z}^2} x(k, l) w_{k-i,l-j} \tag{3.7}$$

This equation effectively states that the output of a convolutional layer (at position i, j) is given by the inner product of the input x with a convolutional kernel shifted (translated) by $(-i, -j)$. The inverse (negative) is picked up due to the fact that we

are writing the shift in terms of the filter $\boldsymbol{w}$, rather than the signal x. In the language of group theory, this shift is written as action of the group element $g_{i,j}$ on the filter: $g_{i,j}^{-1} \cdot \boldsymbol{w}_{k,l} = \boldsymbol{w}_{k-i,l-j}$. If we simply substitute our notation for abstract groups above, we get:

$$[\boldsymbol{w} \star \boldsymbol{x}](g) = \sum_{h \in \mathcal{G}} x(h) w_{g^{-1} \cdot h} \tag{3.8}$$

This is precisely the equation for a group-convolution developed in the seminal work of T. Cohen and Max Welling (2016a).[2] The importance of the work of Cohen and Welling was that they showed that indeed using this abstraction, the equivariance relation ($f(\tau_g[\boldsymbol{x}]) = \Gamma_g[f(\boldsymbol{x})]$) still holds for these G-convolutional neural networks when the translation group is replaced with other abstract groups, such as rotation, roto-translation, and mirroring.

Although we refer readers to the original work for a rigorous understanding of group equivariant neural networks, in what follows we will give a sketch of how equivariance can be shown to hold for these networks. Again, we will start with translation and attempt to generalize to other groups. To begin, consider how translating an image x by an offset (p, q) changes the output a convolutional layer. Denoting this translated image as $x'(i, j) = x(i + p, j + q)$ we have:

$$[\boldsymbol{w} \star \boldsymbol{x}'](i, j) = \sum_{(k,l) \in \mathbb{Z}^2} x(k + p, l + q) w_{k-i,l-j} \tag{3.9}$$

Leveraging the closure of the group $\mathcal{G} = (\mathbb{Z}^2, +)$, we see that we can substitute the sum indices (k, l) with $(k', l') = (k + p, l + q)$ without changing the sum. In doing so, we see that we get:

$$[\boldsymbol{w} \star \boldsymbol{x}'](i, j) = \sum_{(k',l') \in \mathbb{Z}^2} x(k', l') w_{k'-(i+q),l'-(j+p)} = [\boldsymbol{w} \star \boldsymbol{x}](i + q, j + p) \tag{3.10}$$

In this simple example, we have shown that convolution of a translated image $\boldsymbol{x}'$ is equivalent to convolving the original image $\boldsymbol{x}$, and then translating the output. This shows the convolution is 'equivariant' to the translation group, and the 'representation' of the translation operator in the output space Γ_g is actually simply another translation: $(i, j) \to (i + p, j + q)$.

Let us then abstract this a bit to fit with Equation 3.8. We see that through equivalent analysis, in the general group setting, we can substitute h with $h' := g_1 \cdot h$, and again

[2] In the original work, the authors have included a definition which supports multiple input and output channels, however for the sake of cleanliness we have omitted the extra sum and indices needed for this in this overview. A full definition including channels can be found in Chapter 9.

pull the group element out of the sum and into the index:

$$[\boldsymbol{w} \star \boldsymbol{x}'](g_0) = \sum_{h' \in \mathcal{G}} x(h') w_{(g_1 \cdot g_0)^{-1} \cdot h'} = [\boldsymbol{w} \star \boldsymbol{x}](g_1 \cdot g_0) \tag{3.11}$$

As a result, we see that transforming the input by g_1 and then convolving is equivalent to convolving the untransformed input and then applying g_1 to the output.

In going through this example, we also begin to gain some intuition for how we would build a general group equivariant neural network. Rather than having filters defined over just space (the translation group), our filters now must be defined over the full group. In other words, we must have a transformed copy of each of our filters for each of the group elements g (i.e. each rotation angle or scale). Similarly, the output of the network will now have a separate output value for each element of the group, given by applying the transformed filter to the input. When a transformation is then observed to be applied to the input, one sees that the corresponding transformed filter will be selectively responsive. In this way, the output of the network is *structured* with respect to the given group transformation – the designer of the network knows how the output will transform for a given transformation of the input. It is precisely these transformed sets of filters and ensuing related outputs that define the 'equivariant capsules' we alluded to in Section 2.1 in reference to the idea of 'hypercolumns'.

Despite the tremendous success of equivariance as a guiding principle for neural network architectural design, it is still not known how to construct networks with equivariance with respect to many natural transformations, such as lighting or perspective shift, due to their complex non-group structure. How then might natural systems handle the dramatic changes in lighting from day to day without getting confused? This will be one of the motivating questions for our work in this thesis. Precisely, in this work we hypothesize that the natural structure that we observe in the brain (i.e. spatial and spatio-temporal organization) may facilitate the learning of (approximately) equivariant architectures.

3.3. Training Artificial Neural Networks

The most successful and therefore popular method for training today's neural networks is the backpropagation algorithm. For a given weight/parameter matrix $\boldsymbol{W}_l$ of an MLP or other artificial neural network, and a loss function $\mathcal{L}$, the backpropagation algorithm leverages the chain rule of calculus to compute the gradient of the loss for successively deeper layers of the network, starting from the output. Explicitly,

denoting the output of layer l as $\boldsymbol{a}_l$, the gradient of $\mathcal{L}$ with respect to $\boldsymbol{W}_l$ is:

$$\nabla_{\boldsymbol{W}_l}\mathcal{L} = \left(\sigma'(\boldsymbol{a}_l) \odot \boldsymbol{W}_{l+1}^T \cdots \sigma'(\boldsymbol{a}_{L-1}) \odot \boldsymbol{W}_L^T \cdot \sigma'(\boldsymbol{a}_L) \odot \nabla_{\boldsymbol{a}_L}\mathcal{L}\right) \boldsymbol{a}_{l-1}^T \tag{3.12}$$

Since we know that the gradient of a function is a vector which points in the direction of steepest increase, the vector $-\nabla_{\boldsymbol{W}_l}\mathcal{L}$ will precisely contain the updates necessary to optimally decrease our loss. Explicitly then, using this gradient descent procedure, the weights of the model are updated as: $\boldsymbol{W}_l^{(new)} = \boldsymbol{W}_l^{(old)} - \lambda\nabla_{\boldsymbol{W}_l}\mathcal{L}$ where λ is referred to the learning rate of our optimization procedure and must be tuned to ensure learning proceeds optimally. Typically, the gradient of the loss is not computed with respect to loss on the full dataset as would be required to compute the exact gradient, but rather with respect to a small batch of data. This means that our optimization procedure will no longer follow the exact gradient of the loss, but rather it will have an element of stochasticity in the random selection of the batch, thereby earning the name *stochastic gradient descent*.

In natural systems, it is widely accepted that an analogous learning rule to back-propagation is implausible for biological neurons for a multitude of reasons. The first and perhaps most well known is the 'weight transport problem' (Lillicrap, Cownden, et al., 2014). This problem arises from the fact that the terms required to pass the error between successive layers are in the form of transposed versions of the forward parameters (i.e. the $\boldsymbol{W}_i^T$ terms in Equation 3.12). Although feedback connections are widely observed throughout the brain, there is no known mechanism by which these connections could be made to be exact transposed copies of their 'feedforward' counterparts. Such a mechanism would be equivalent to some form of 'transportation' copying the weights and maintaining this equivalence as weights are updated during learning. A second implausible property of the backpropagation algorithm is the fact that it requires the evaluation of the activation function at the exact point where the forward pass was evaluated. This is given by the term $\sigma'(\boldsymbol{a}_i)$, of Equation 3.12. It seems very unlikely indeed that the inverse connections are able to implement the exact gradient of the forward nonlinearity in a manner which is modulated by the forward propagated signal (D.-H. Lee et al., 2015). These are just two examples of the many discrepancies that researchers have aimed to overcome with more biological alternatives in recent years.

Despite these differences in the training procedure, researchers have still found backpropagation and stochastic gradient descent to be valuable methods for learning the weights of deep neural network architectures which ultimately end up resembling

natural neural networks on a variety of fronts (Daniel L. K. Yamins et al., 2014; Cadieu et al., 2014; Conwell et al., 2023). In this work, we will therefore train all networks with backpropagation, hoping to understand the fundamentals of neural representations first and foremost, and leave the question of how local gradient-free learning may be accomplished to future work.

Supervised Learning

We note that an important part of the proceeding section which is left undefined is the requirement for some sort of supervision or target signal y. Depending on the task, this supervisory signal will vary to match the task requirements. For example, for classification, y is defined as the class label of the example and $\mathcal{L}$ is often defined as a form of cross entropy between the predicted label distribution and the true label distribution. For something such as semantic image segmentation, the supervision signal y will be a pixel-wise mask which labels each individual pixel in the image as falling into one of the desired classes, and a pixel-wise cross entropy loss is again applied. However, these are only a few limited examples of the virtually limitless space of possible supervision signals for training neural network models. In our work we will make use of a few such supervised tasks to evaluate our models. However, for a larger section of the work, we will focus on tasks which have no explicit label but are instead focused on learning valuable representations from the data itself.

Self-Supervised Learning

One framework which has recently increased in popularity for learning representations from data without human labels is that of self-supervised learning. Many self-supervised learning (SSL) techniques can be colloquially defined as representation learning algorithms which extract approximate supervision signals directly from the input data itself (LeCun and Misra, 2021). In practice, this supervision signal is often obtained by performing symmetry transformations of the input with respect to task-relevant information, meaning the transformations leave task-relevant information unchanged, while altering task-irrelevant information. Numerous theoretical and empirical works have shown that by combining such symmetry transformations with specific contrastive learning objectives, powerful lower dimensional representations can be learned which support linear-separability (Wang, Q. Zhang, et al., 2022; J. D. Lee et al., 2021; Yuandong Tian, Yu, et al., 2020; Arora et al., 2019;

Tosh et al., 2020), identifiability of generative factors (Kügelgen et al., 2021; Tsai, Wu, et al., 2020; Federici et al., 2020; Ji et al., 2021), and reduced sample complexity (Grill et al., 2020; T. Chen et al., 2020). At a high level, these contrastive objectives can be seen as encouraging the model to produce similar representations for two symmetrically transformed versions of the same image, while encouraging dissimilarity of representations for two entirely separate images (which share no symmetries). In Chapter 9 we show how a more biologically plausible form of self-supervised learning can be performed without the need for explicit input transformations through the use of the symmetries embedded in structured equivariant neural networks themselves.

Unsupervised Learning

For all other chapters we will mainly focus on unsupervised learning rules derived from the frameworks of auto-encoding and probabilistic generative modeling. Our models will therefore still use stochastic gradient descent for learning model parameters, but will not rely on explicit human annotations as 'labels'. While there do exist a variety of exciting gradient-free learning rules, including local learning rules such as those based on hebbian learning (Hebb, 1949; Journé et al., 2022), and approximations to the gradient such as target propagation (Bengio, 2014) and synthetic gradients (Jaderberg et al., 2017), we will not address them directly in this thesis.

In an auto-encoder, the goal of training is to learn an 'encoder' mapping from the input to a hidden state (often of reduced dimensionality), and then similarly learn a 'decoder' to reconstruct the input. In reference to Equation 3.1 above, an auto-encoder could be defined as having an encoder composed of the first $E = \lfloor \frac{L}{2} \rfloor$ layers, taking the activation of f_E as the internal hidden representation, and subsequently defining the final $D = L - E$ layers as the decoder. In such a model, the target is then defined to be equal to the input $y = x$, and the loss function is defined to measure the distance (often L2 norm) between the output of the model (often called the 'reconstruction' $\hat{\boldsymbol{x}}$) and the original input x.

As an alternative, the framework of probabilistic generative modeling can be seen as an attempt to specify a model of the *data generating process* itself, and in doing so, a procedure is usually defined to be able to invert this process and thereby access the unobserved 'latent variables' which correspond to each input. As an example, consider a dataset consisting simply of the weights of all cars in a given

country. If our goal is to model such a dataset, it may make sense to imagine that there are unobserved 'latent variables' such as the type of vehicle (sedan vs. truck) which are largely responsible for the variety in the observed weight distribution, and the remaining differences are best modeled by noise. The data generating process for a single data-point in our dataset could then be seen as first randomly sampling the type of vehicle $\mathbf{t}$ from a prior distribution $\mathbf{t} \sim p_{\mathbf{T}}(\mathbf{t})$ which defines the common types of cars in the given country, then simply taking the average value of the weight $\mathbf{w}$ for this car type and adding a small amount of noise, i.e. $\mathbf{w} \sim p_{\mathrm{W}|\mathbf{T}}(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}; \mu_\theta(\mathbf{t}), \sigma_\theta(\mathbf{t}))$. The goal of training is then to learn the parameters θ such that the marginal distribution of the model $p_\theta(\mathbf{w})$ matches that of the observed data. Unfortunately, computing this marginal likelihood exactly is generally intractable as it requires computing an integral over all possible states of the latent variable $\mathbf{t}$ (e.g. $p_\theta(\mathbf{w}) = \int_t p_{\mathrm{W}|\mathbf{T}}(\mathbf{w}|\mathbf{t})dt$). One popular ways around this intractability is through the framework of variational inference and the ensuing variational autoencoders (Kingma and Max Welling, 2014). In this framework, the true unknown posterior $p_{\mathrm{T}|\mathbf{W}}(\mathbf{t}|\mathbf{w})$ is approximated by a function $q_\phi(\mathbf{t}|\mathbf{w})$, and the parameters ϕ of this distribution are optimized to make this approximate posterior as close to the true posterior as possible. Of course, since the true posterior is unknown, this is again not possible to do directly. Instead, the approximate posterior is used to derive a lower bound of the likelihood of the data known as the Evidence Lower Bound (ELBO):

$$\log p_\theta(\mathbf{w}) \geq \mathbb{E}_{\mathbf{t}\sim q_\phi(\mathbf{t}|\mathbf{w})}\left[\log p_{\mathrm{W}|\mathbf{T}}(\mathbf{w}|\mathbf{t})\right] - D_{KL}(q_\phi(\mathbf{t}|\mathbf{w})||p_{\mathbf{T}}(\mathbf{t})) \tag{3.13}$$

Such models can be seen as defining an auto-encoder, with encoder q_ϕ and decoder p_θ, which additionally have a penalty on the divergence of the code in latent space from some pre-defined prior distribution. In practice, this formalization allows for a principled method of model comparison through the likelihood of the data, as well as the introduction of additional inductive biases through the choice of prior and posterior distributions. In our work we will demonstrate how this additional flexibility allows for the creation of inductive biases which match those observed in natural intelligence.

Part I

Spatial Structure

Chapter 4

TOPOGRAPHIC ORGANIZATION

4.1. Introduction

Topographic organization in the brain describes the observation that nearby neurons on the cortical surface tend to have more strongly correlated activations than spatially distant neurons. From the simple orientation of lines (David H. Hubel and Torsten N. Wiesel, 1974b) to the complex semantics of natural language (Huth et al., 2016), organization of cortical activity is observed for a diversity of stimuli and across a range of species. Perhaps most well known are the local regions of category selectivity found through the higher areas of the visual stream. These regions have been measured to respond preferentially to specific stimuli from their respective categories when compared with a set of alternative control images. Their existence has been measured across a diversity of species (Kanwisher, McDermott, et al., 1997; Tsao et al., 2006; Nasr et al., 2011), directly through fMRI and neural recordings (Pinsk et al., 2005a), and more indirectly through observational studies of patients with localized cortical damage (Moro et al., 2008). Examples of category-selective areas in the visual stream include the Fuisform Face Area (FFA) (Kanwisher, McDermott, et al., 1997), the Parahippocampal Place Area (PPA) (Epstein and Kanwisher, 1998; Nasr et al., 2011), and the Extrastriate Body Area (EBA) (Peelen and Downing, 2005) which respond selectively to faces, places, and bodies respectively. However, the extent of category-selectivity does not stop at such basic categories. Instead, selective maps have been observed for both more abstract 'superordinate' categories, such as animacy versus inanimacy (Haxby, Guntupalli, et al., 2011; Konkle and Oliva, 2012), as well as for more fine-grained 'subordinate' categories such as human-faces versus animal-faces (Haxby, Gobbini, et al., 2001). These maps are seen to be superimposed on one-another such that the same cortical region expresses selectivity simultaneously to animate objects and human-faces, while other spatially disjoint regions are simultaneously selective to inanimacy and 'places' (images of scenes). Such overlapping maps have been interpreted by some researchers as nested hierarchies of increasingly abstract categories, potentially serving to increase the speed and efficiency of classification (Grill-Spector and K. S. Weiner, 2014). In interpreting these observations, one may naturally wonder as to

the origins of such localized specialization. From the current literature, the driving factors can roughly be divided into two potentially complimentary categories: anatomical, and information theoretic.

Anatomically, the arrangement and properties of different cell bodies can be observed to vary slightly in different regions of the cortex in loose alignment with category selectivity (K. S. Weiner et al., 2014; Caspers et al., 2013; Saygin et al., 2012), possibly serving as an innate blueprint for specialization. In the same category, the principle of 'wiring length minimization' (Koulakov and Chklovskii, 2001; Essen, 1997) posits that evolutionary pressure has encouraged the brain to reduce the cumulative length of neural connections in order to reduce the costs associated with the volume, building, maintenance, and use of such connections. Computational models which attempt to integrate such wiring length constraints (H. Lee et al., 2020; Y. Zhang et al., 2021; Blauch et al., 2021) have recently been observed to yield localized category selectivity such as 'face patches' similar to those of macaque monkeys.

From the information theoretic perspective, one potential explanation for the emergence of topographic organization is provided by the principle of redundancy reduction (Barlow et al., 1961). Simply, the principle states that an optimal coding scheme is one which minimizes the transmission of redundant information. Applied to neural systems, this describes the ideal network as one which has statistically maximally independant activations – yielding a form of specialization. This idea served as the impetus for computational frameworks such as Sparse Coding (Olshausen and Field, 1997) and Independant Component Analysis (ICA) (Bell and Terrence J. Sejnowski, 1995; Comon, 1994; Aapo Hyvärinen, 1998; Aapo Hyvärinen and Oja, 2000). Interestingly, however, further work showed that features learned by linear ICA models were not entirely independant, but indeed contained correlation of higher order statistics (such as correlation between absolute values). For example, along edges of an image, linear-ICA components (e.g. gabor filters) still activate in clusters even though the sign of their activity is unpredictable (Portilla et al., 2003; Wainwright and E. P. Simoncelli, 2000). In response, researchers proposed a more efficient code could be achieved by modeling these residual dependencies with a hierarchical topographic extension to ICA (Aapo Hyvärinen, Patrik O Hoyer, et al., 2001; Aapo Hyvärinen and Patrik O. Hoyer, 2001), separating out the higher order 'variance generating' variables, and combining them locally to form topographically organized latent variables. Such a framework shares a striking resemblance to

models of divisive normalization (Lyu and E. P. Simoncelli, 2009a) (another known neurobiological motif), but inversely formulated as a generative model. Ultimately, the features learned by such models were reminiscent of pinwheel structures observed in V1, encouraging multiple comparisons with topographic organization in the biological visual system (Aapo Hyvärinen, Hurri, et al., 2009; Aapo Hyvärinen and Patrik O. Hoyer, 2001; Ma and L. Zhang, 2008).

Due to the nature of the learning algorithms used in these early frameworks however, they were restricted to learning linear generative models and therefore were unable to extend to the complex and varied forms of topography we see in the brain. It has therefore remained unclear if these types of redundancy reduction arguments are sufficient to explain topographic organization at all levels of the cortical hierarchy, such as the category selective face, body, and place areas described above. In this chapter, we aim to close this gap by introducing a deep nonlinear generative model capable of modeling such topographic organization of more abstract concepts. To accomplish this, we leverage the framework of Variational Autoencoders, and introduce a new generative component which facilitates the same 'group-sparse' priors that were developed decades earlier for Topographic ICA.

4.2. Related Work

The history of statistical models upon which this work builds is vast, including sparse coding (Olshausen and Field, 1997), Independant Component Analysis (ICA) (Bell and Terrence J. Sejnowski, 1995; Comon, 1994; Aapo Hyvärinen and Oja, 2000), the Helmholtz Machine (Dayan et al., 1995), and of course Variational Autoencoders (Kingma and Max Welling, 2014). Most related to this work are topographic generative models including Generative Topographic Maps (Bishop et al., 1997), Bubbles (A. Hyvärinen et al., 2004), Topographic ICA (Aapo Hyvärinen, Patrik O Hoyer, et al., 2001), and the Topographic Product of Student's-t models (S. K. Osindero, 2004; Max Welling et al., 2003). Although our work is not the first to combine Student's-t distributions and variational inference (Boenninghoff et al., 2020), it is the first to provide an efficient method to do so for Topographic Student's-t distributions.

Recently, a number of models of topographic organization in the visual system have been developed leveraging modern deep neural networks. Y. Zhang et al. (2021) demonstrated category-selective regions, as well as a nested spatial hierarchy of selectivity, through the use of self-organizing maps (SOMs). Due to the challenges

with scaling SOMs, the inputs were dimensionality-reduced with PCA, limiting the applicability of the algorithm to arbitrary neural network architectures. More recently, Doshi and Konkle (2022) further showed that self organizing maps could be used on full-dimensionality network activations to reproduce forms of topographic organization from large-scale to mid-level feature tuning. Concurrently with our work, Blauch et al. (2021) developed the Interactive Topographic Network (ITN), inducing local correlation through locally-biased excitatory feedforward connections in a biologically-constrained model. Most related to our work, the TDANN of H. Lee et al. (2020) incorporated a biologically derived proxy for wiring length cost into the fully connected layers of a supervised Alexnet model (Krizhevsky, Sutskever, et al., 2012), and similarly demonstrated emergent localized category-selectivity. Our model differs from these in that it explicitly formulates a properly normalized density over the input data with topographic organization originating as a prior over latent the variables – thereby unifying feature extraction and topographic organization into a single training objective: maximization of the data likelihood. Interestingly, the model presented in this chapter organizes activity based on the same statistical property (local correlation) as the wiring length proxies developed by H. Lee et al. (2020), but from a generative modeling perspective, thereby encouraging unsupervised representation learning and topographic organization in through a single unified framework.

Finally, there are a number of recent empirical studies which motivated our investigation of category selectivity from a purely information theoretic perspective. Specifically, the work of Dobs et al. (2021) demonstrated that sufficiently deep convolutional neural networks naturally learn distinct and largely separate sets of features for certain domains such as faces and objects. In this work, the authors showed that feature maps in the later layers of deep convolutional neural networks can be effectively segregated into object and face features such that lesioning one set of feature maps does not significantly impact performance of the network on classification of the other data domain. Such experiments, and others (Bakhtiari et al., 2021; Konkle and Alvarez, 2021), suggest that the specialization of neurons may simply be an optimal code for representing the natural statistics of the underlying data when given a sufficiently powerful feature extractor, and therefore prompted us to investigate if this specialization combined with topographic generative models may yield localized clusters of category selectivity.

4.3. Background

The model presented in this chapter extends the existing set of topographic generative models to allow integration with deep neural network function approximators. In this section we provide a brief background of this class of models.

Topographic Generative models

Inspired by Topographic ICA, the class of topographic generative models can be understood as generative models where the joint distribution over latent variables does not factorize into entirely independent factors, as is commonly done in ICA or variational autoencoders, but instead has a more complex 'local' correlation structure. The locality is defined by arranging the latent variables into an n-dimensional lattice or grid, and organizing variables such that those which are closer together on this grid have greater correlation of activities than those which are further apart. In the related literature, activations which are nearby in this grid are defined to have higher-order correlation, e.g. correlations of squared activations (aka 'energy'), asserting that all first order correlations are removed by the initial ICA de-mixing matrix.

Such generative models can be seen as hierarchical generative models where there exist higher level independent 'variance generating' variables $\mathbf{V}$ which are combined locally to generate the variances $\boldsymbol{\sigma} = \phi(\mathbf{WV})$ of the lower level topographic variables $\mathbf{T} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}^2\mathbf{I})$, for an appropriate non-linearity ϕ. The variables $\mathbf{T}$ are thus independent conditioned on $\boldsymbol{\sigma}$. Other related models which can be described under this umbrella include *Independent Subspace Analysis* (ISA) (Aapo Hyvärinen and P. Hoyer, 2000) where all variables within a predefined subspace (or 'capsule') share a common variance, and '*temporally coherent*' models (Hurri and Aapo Hyvärinen, 2003) where the energy of a given variable between time steps is correlated by extending the topographic neighborhoods over the time dimension (A. Hyvärinen et al., 2004). The topographic latent variable $\mathbf{T}$ can additionally be described as an instance of a Gaussian scale mixture (GSM). GSMs have previously been used to model the observed non-Gaussian dependencies between coefficients of steerable wavelet pyramids (and are interestingly also equivariant to translation & rotation) (Portilla et al., 2003; Wainwright and E. P. Simoncelli, 2000; Wainwright, E. P. Simoncelli, and Willsky, 2001).

4.4. The Generative Model

The generative model proposed in this chapter is based on the Topographic Product of Student's-t (TPoT) model as developed in (S. K. Osindero, 2004; Max Welling et al., 2003). In the following, we will show how a TPoT random variable can be constructed from a set of independent univariate standard normal random variables, enabling efficient inference through the framework of variational autoencoders.

The Product of Student's-t Model

We assume that that our observed data is generated by a latent variable model where the joint distribution over observed and latent variables $\mathbf{x}$ and $\mathbf{t}$ factorizes into the product of the conditional and the prior. The prior distribution $p_{\mathbf{T}}(\mathbf{t})$ is assumed to be a Topographic Product of Student's-t (TPoT) distribution, and we parameterize the conditional distribution with a flexible function approximator:

$$p_{\mathbf{X},\mathbf{T}}(\mathbf{x},\mathbf{t}) = p_{\mathbf{X}|\mathbf{T}}(\mathbf{x}|\mathbf{t})p_{\mathbf{T}}(\mathbf{t}) \qquad p_{\mathbf{X}|\mathbf{T}}(\mathbf{x}|\mathbf{t}) = p_\theta(\mathbf{x}|g_\theta(\mathbf{t})) \qquad p_{\mathbf{T}}(\mathbf{t}) = \mathrm{TPoT}(\mathbf{t};\nu) \tag{4.1}$$

The goal of training is thus to learn the parameters θ such that the marginal distribution of the model $p_\theta(\mathbf{x})$ matches that of the observed data. Unfortunately, the marginal likelihood is generally intractable except for all but the simplest choices of g_θ and $p_{\mathbf{T}}$ (S. Osindero et al., 2006). Prior work has therefore resorted to techniques such as contrastive divergence with Gibbs sampling (Max Welling et al., 2003) to train TPoT models as energy based models. In the following, we instead demonstrate how TPoT variables can be constructed as a deterministic function of Gaussian random variables, enabling the use of variational inference and efficient maximization of the likelihood through the evidence lower bound (ELBO).

Constructing the Product of Student's-t Distribution

First, note a univariate Student's-t random variable T with ν degrees of freedom can be defined as:

$$T = \frac{Z}{\sqrt{\frac{1}{\nu}\sum_i^\nu U_i^2}} \quad \text{with} \quad Z, U_i \sim \mathcal{N}(0,1) \;\; \forall i \tag{4.2}$$

Where Z and $\{U_i\}_{i=1}^\nu$ are independent standard normal random variables. If $\mathbf{T}$ is a multidimensional Student's-t random variable, composed of independent Z_i and U_i,

then $\mathbf{T} \sim \text{PoT}(\nu)$, i.e.:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu}\sum_{i=1}^{\nu} U_i^2}}, \quad \frac{Z_2}{\sqrt{\frac{1}{\nu}\sum_{i=\nu+1}^{2\cdot\nu} U_i^2}}, \quad \ldots \quad \frac{Z_n}{\sqrt{\frac{1}{\nu}\sum_{i=(n-1)\cdot\nu+1}^{n\cdot\nu} U_i^2}} \right] \sim \text{PoT}(\nu) \tag{4.3}$$

Note that the Student's-t variable T is large when most of the $\{U_i\}_i$ in its set are small. We can therefore think of the $\{U_i\}_i$ as constraint violations rather than pattern matches: if the input matches all constraints $U_i \approx 0$, the corresponding T variables will activate (see (Geoffrey E. Hinton and Teh, 2001) for further discussion on the relative benefits of a constraint violation framework compared with standard 'feature-detector' frameworks.).

Introducing Topography

To make the PoT distribution topographic, we strive to correlate the scales of T_j which are 'nearby' in our topographic layout. One way to accomplish this is by *sharing* some U_i-variables between neighboring T_j's. Formally, we define overlapping neighborhoods $\mathsf{N}(j)$ for each variable T_j and write:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu}\sum_{i\in\mathsf{N}(1)} U_i^2}}, \quad \frac{Z_2}{\sqrt{\frac{1}{\nu}\sum_{i\in\mathsf{N}(2)} U_i^2}}, \quad \ldots \quad \frac{Z_n}{\sqrt{\frac{1}{\nu}\sum_{i\in\mathsf{N}(n)} U_i^2}} \right] \sim \text{TPoT}(\nu) \tag{4.4}$$

With some abuse of notation, if we define $\mathbf{W}$ to be the adjacency matrix which defines our neighborhood structure, $\mathbf{U}$ and $\mathbf{Z}$ to be the vectors of random variables U_i and Z_j, we can write the above succinctly as:

$$\mathbf{T} = \left[\frac{Z_1}{\sqrt{\frac{1}{\nu}W_1\mathbf{U}^2}}, \quad \frac{Z_2}{\sqrt{\frac{1}{\nu}W_2\mathbf{U}^2}}, \quad \ldots \quad \frac{Z_n}{\sqrt{\frac{1}{\nu}W_n\mathbf{U}^2}} \right] = \frac{\mathbf{Z}}{\sqrt{\frac{1}{\nu}\mathbf{W}\mathbf{U}^2}} \sim \text{TPoT}(\nu) \tag{4.5}$$

Due to non-linearities such as ReLUs which may alter input distributions, it is beneficial to allow the Z variables to model the mean and scale. We found this can be achieved with the following parameterization: $\mathbf{T} = \frac{\mathbf{Z}-\mu}{\sigma\sqrt{1/\nu\mathbf{W}\mathbf{U}^2}}$. In practice, we found that $\sigma = \sqrt{\nu}$ often works well, finally yielding:

$$\mathbf{T} = \frac{\mathbf{Z} - \mu}{\sqrt{\mathbf{W}\mathbf{U}^2}} \tag{4.6}$$

Given this construction, we observe that the TPoT generative model can instead be viewed as a latent variable model where all random variables are Gaussian and

the construction of $\mathbf{T}$ in Equation 4.6 is the first layer of the generative 'decoder': $g_\theta(\mathbf{t}) = g_\theta(\mathbf{u}, \mathbf{z})$. In the next section we then leverage this interpretation to show how an approximate posterior for the latent variables $\mathbf{Z}$ and $\mathbf{U}$ can be trained through variational inference.

4.5. The Topographic VAE

To train the parameters of the generative model θ, we use the above formulation to parameterize an approximate posterior for $\mathbf{t}$ in terms of a deterministic transformation of approximate posteriors over simpler Gaussian latent variables $\mathbf{u}$ and $\mathbf{z}$. Explicitly:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mathbf{z}; \mu_\phi(\mathbf{x}), \sigma_\phi(\mathbf{x})\mathbf{I}\big) \qquad q_\gamma(\mathbf{u}|\mathbf{x}) = \mathcal{N}\big(\mathbf{u}; \mu_\gamma(\mathbf{x}), \sigma_\gamma(\mathbf{x})\mathbf{I}\big) \tag{4.7}$$

$$\mathbf{t} = \frac{\mathbf{z} - \mu}{\sqrt{\mathbf{W}\mathbf{u}}} \qquad p_\theta(\mathbf{x}|g_\theta(\mathbf{t})) = p_\theta(\mathbf{x}|g_\theta(\mathbf{z}, \mathbf{u})) \tag{4.8}$$

The parameters θ, ϕ, γ and μ are then optimized to maximize the likelihood of the data through the Evidence Lower Bound (ELBO):

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})q_\gamma(\mathbf{u}|\mathbf{x})}\left(\log p_\theta(\mathbf{x}|g_\theta(\mathbf{t})) - D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\mathbf{Z}(\mathbf{z})] - D_{KL}[q_\gamma(\mathbf{u}|\mathbf{x})||p_\mathbf{U}(\mathbf{u})]\right) \tag{4.9}$$

MNIST Validation

To validate the TVAE is capable of learning topographically organized representations with deep neural networks, we first perform experiments training a Topographic VAE as in Equations 4.7 and 4.8 to maximize Equation 4.9 on the simple MNIST dataset composed of grayscale images of handwritten digits (LeCun and Cortes, 2010). We use minimal 3-layer MLPs for the encoders and decoders, and fix $\mathbf{W}$ such that globally the latent variables are arranged in a grid on a 2-dimensional torus (a single capsule) and locally $\mathbf{W}$ sums over 5x5 2D groups of variables. In this setting, $\mathbf{W}$ can be easily implemented as 2D convolution with a 5x5 kernel of 1's, stride 1, and cyclic padding.

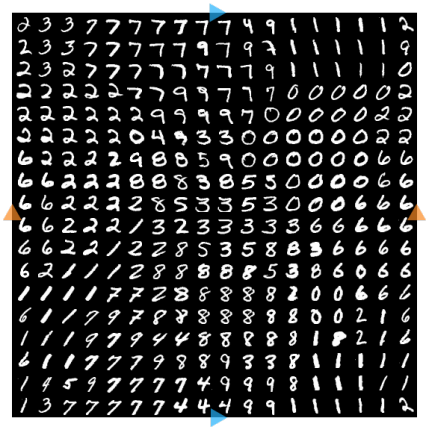


Figure 4.1: Maximum activating images for a Topographic VAE trained with a 2D torus topography on MNIST.

As an intuitive computationally simple metric of spatial selectivity, we use a visualization of the maximum activating images from our dataset for each neuron. In detail, for each of our neurons organized in a 2-D regular grid, we plot the image from the dataset which produces the maximum absolute activation value of that neuron. In doing so, we can see qualitative similarities between images, and thus the selectivities of neurons, which would be difficult to capture with pure statistical or category based selectivity measures alone. In Figure 4.1 we see that training the model indeed yields a 2D topographic organization of higher level features. We show the maximum activating image for each final layer neuron of the capsule, plotted as a flattened torus and see that the neurons become arranged according to class, orientation, width, and other learned features.

4.6. Methods

In this section we explain the datasets, evaluation metrics, and model architectures used to validate the Topographic VAE's emergent higher level topographic organization of category selectivity when trained on more complex natural images. Additionally, we provide details on the baseline we have re-implemented for these experiments: the TDANN of H. Lee et al. (2020).

Evaluation

Following prior computational work (H. Lee et al., 2020; Y. Zhang et al., 2021) and fMRI studies (Aparicio et al., 2016), we will make use of Cohen's d metric (J. Cohen, 1988; Sawilowsky, 2009), a measure of standardized difference of two means, as our selectivity metric. Given the means $\bar{m}_1$ & $\bar{m}_2$ and standard deviations σ_1 & σ_2 of two sets of data, the d metric is given as:

$$d = \frac{\bar{m}_1 - \bar{m}_2}{\sqrt{\frac{1}{2}\left(\sigma_1^2 + \sigma_2^2\right)}} \tag{4.10}$$

This value is unitless and can be seen as expressing the difference between two means in terms of units of 'pooled variability'. In this work, the mean $\bar{m}_1$ corresponds to the mean activation of a single neuron computed across an entire dataset of class-specific target images (e.g. faces), while $\bar{m}_2$ is the mean activation of the same neuron across a dataset of control images which do not contain this class.

Datasets

Following our simple validation on MNIST, we perform a suite of experiments to study if our model is able to learn higher order category selectivity. To accomplish this, the training diet of the model is crucially important (Conwell et al., 2023). Therefore, for this set of experiments involving both the TDANN and TVAE, we use a dataset composed of a combination of the ImageNet 2012 (Russakovsky et al., 2015) and Labeled Faces in the Wild (LFW) datasets (Huang et al., 2007), following H. Lee et al. (2020). The TDANN was trained to classify the 1000 distinct image classes from ImageNet, plus one generic face class encompassing all of LFW. The TVAE used no such class labels. To measure the category selectivity of the models, the primary test face dataset used in Figures 4.2, 4.3, & 4.5 was a ~25,000 image subset of VGGface2 (Cao et al., 2018). The control 'object' dataset for Figures 4.2 & 4.5 was composed of 25,000 images from the validation set of ImageNet. To measure selectivity to body parts and places in Figure 4.3, we created a 'body' dataset composed of headless body images (Clemons, 2018) and hands (Afifi, 2019), and used the Place365 dataset (Zhou et al., 2017) for places. In Figure 4.3, the 'control' set used for each class was defined to be the compliment of the test set, i.e. all other datasets besides the target category of interest.

Models

On the ImageNet dataset, all models are trained on top of features extracted by the final convolutional layer of a pre-trained Alexnet model (Krizhevsky, Sutskever, et al., 2012; Paszke et al., 2019). The Alexnet architecture was chosen to match the setup from H. Lee et al. (2020) and Y. Zhang et al. (2021), and has further been shown to have remarkable similarities to hierarchical processing in the human visual stream (Daniel L K Yamins and DiCarlo, 2016; Cichy et al., 2016; Güçlü and Gerven, 2015). For the TVAE, we randomly initialize and train a single linear layer encoder and decoder with 4096 output neurons, arranged in a 64x64 grid with circular boundary conditions to avoid edge effects. For the TDANN, we randomly initialize and train all three fully connected layers of Alexnet, imposing the spatial correlation loss over both 'FC6' and 'FC7'. The exact form of the spatial correlation loss used for training the TDANN in this chapter is given as:

$$\text{SpatialCorrelationLoss}(\mathbf{z}) = \sum_{i}^{n} \sum_{j \neq i}^{n} \left| C_{ij}(\mathbf{z}) - \frac{1}{D_{ij} + 1} \right| \tag{4.11}$$

where $\mathbf{z}$ an n-dimensional vector of activations, C is the normalized cross correlation matrix (e.g. a matrix of Pearson correlation coefficients), and D is a matrix containing the 'cortical distances' in millimeters between all pairs of neurons i and j. For the TDANN, we defined all neurons to be equally spaced in a 2-D grid of 10mm $\times$ 10mm. This resulted in a horizontal and vertical spacing between neurons of 0.15625mm and a diagonal spacing of 0.22097087mm. Unlike the TVAE, the TDANN grid was not defined to have circular boundary conditions in order to match the original model. In the following, all selectivity maps are displayed for 'FC6', following H. Lee et al. (2020). All hyperparameter and training details can be found in the Appendix Section A.2.

4.7. Experiments

Given that our model appears to show category selectivity for simple stimuli such as MNIST digits, we found it natural to wonder if this model may also produce category selective regions for more complex stimuli such as natural images. In the following, we explore the category-selectivity of top-level neurons trained with the Topographic VAE framework on realistic images. We observe that neurons do indeed become category-selective, and that selective neurons tend to group together to form localized category-selective regions for a variety of domains including faces, bodies, and places. We compare these results with a non-topographic baseline (pre-trained Alexnet), and a re-implementation of the TDANN, observing qualitatively similar results. Additionally, following Y. Zhang et al. (2021), we plot selectivity maps to more abstract concepts (such as animacy and real-world size), and observe that such maps overlap in an intuitive manner, suggesting the existence of a nested spatial hierarchy of categories similar to that observed in the brain (Grill-Spector and K. S. Weiner, 2014).

Localized Category-Selectivity

In Figure 4.2, we plot the continuous value of Cohen's d metric for all neurons as arranged in a 2-d grid. The baseline (left) shows the first fully connected layer (FC6) of a pre-trained Alexnet architecture. As expected, the neurons of this model have no defined spatial organization and thus result in a random selectivity map. We note the existence of class-selective neurons is not guaranteed, but their appearance here is in-line with observations from prior work (H. Lee et al., 2020; Raman and Hosoya, 2020). Secondly, we compare our TVAE model (middle) with our re-

implementation of the TDANN (right). We observe that both models demonstrate the emergence of face-selective clusters of comparable size and density. We see that the TVAE framework appears to yield smoother topographic maps, perhaps due to the unified objective function and unsupervised learning rule when compared with the competing supervised classification loss and wiring cost regularization of the TDANN. To validate the robustness and significance of these category selective regions, in Section A.3 of the appendix we plot selectivity maps across four different test face datasets and four random initalizations, observing qualitatively similar clusters all settings.

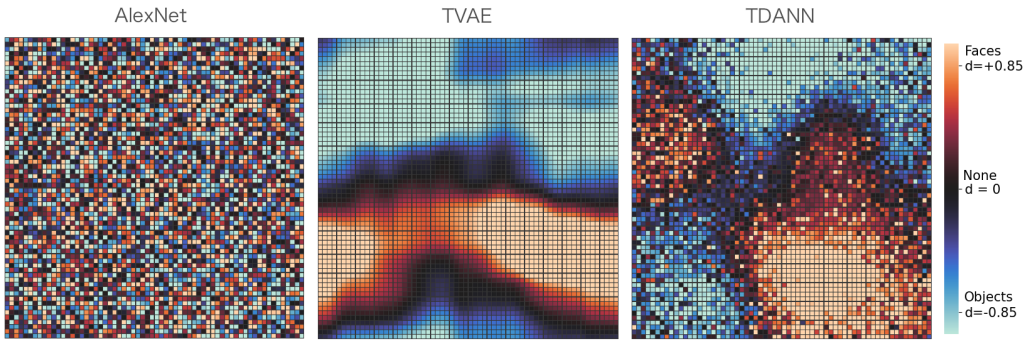


Figure 4.2: Face vs. Object selectivity for a non-topographic baseline, Topographic VAE, and TDANN. We see the TVAE has an emergent face cluster qualitatively similar to that of the TDANN.

Face, Body & Place Clusters

Next, in Figure 4.3, we plot the simultaneous selectivity of neurons in our TVAE model to multiple classes including faces, bodies, and places. To create a map of multi-class selectivity, we follow prior work and threshold the d metric at 0.85, considered a 'strong effect' (Sawilowsky, 2009) and computed to be to be equivalent to a threshold of 0.65 for noisy neural recordings in monkeys (H. Lee et al., 2020). In the plot we observe an overlap of neurons with selectivity to faces and bodies, as seen in prior computational work (H. Lee et al., 2020) and fMRI studies (Pinsk et al., 2005b; K. Weiner and Grill-Spector, 2011). In Figure A.1 of

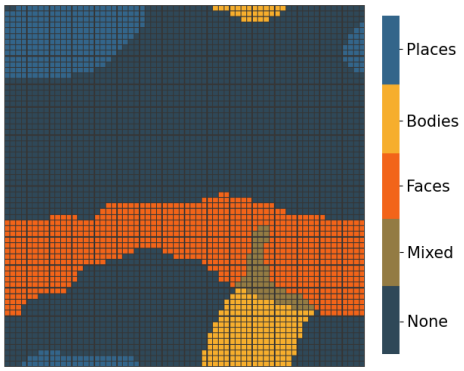


Figure 4.3: TVAE selectivity $d \geq 0.85$

Section A.3, we see that the size and relative placement of these clusters is again consistent across multiple random initalizations.

Impact of Topography on Model Performance

To measure the impact of the imposed topographic organization on the above models, and ensure the learned representations are not degenerate, we compare the model performance of the TDANN and TVAE with their respective non-topographic counterparts. Although the models in this study were not tuned to maximize such performance, we observe that both topographic models perform similarly to their non-topographic counterparts. Specifically the TDANN achieves 40.5% top-1 accuracy on the Imagenet validation set (+ 1 face class) versus the 45.5% top-1 accuracy of an identically trained model without spatial correlation loss. Similarly, a baseline VAE of the same architecture as the TVAE achieves roughly 3.4 bits per dimension (BPD) while the Topographic VAE achieves roughly 3.6 BPD in the same number of iterations. These results appear consistent with the intuition that topographic organization does not prevent learning, but rather acts as an inductive bias on the model, regularizing its performance. In Chapter 5 we will study a situation where this regularization effect is more accurately aligned with the data distribution, specifically in the case of data with known symmetries, and show how it can be beneficial for improved model performance.

Locally Distributed Activations

To understand better how exactly individual images are represented by the TVAE, we present the activation maps corresponding to a single image from an array of classes in Figure 4.4. We see the representation of each image is still distributed, but most strongly activates in the associated category-selective region.

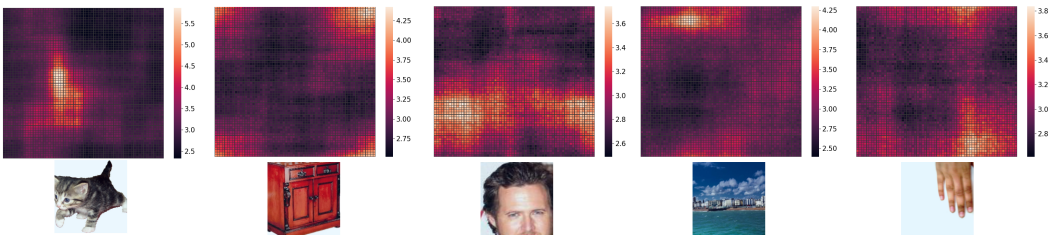


Figure 4.4: Activations for single images. Left to Right: Animate, Inanimate, Faces, Places, & Hands

Nested Spatial Hierarchy of Categories

Following Y. Zhang et al. (2021) we additionally measure the selectivity maps of our TVAE model with respect to more abstract categories such as animacy and real-world object size, obtaining such datasets from the Konkle lab database (Konkle and Oliva, 2012; Konkle and Caramazza, 2013). Specifically, Figure 4.5 shows Cohen's d maps (from -1 to $+1$) for animate versus inanimate objects (top), and for big versus small objects (middle), overlayed on the face versus object map (bottom). At the largest scale, we observe an intuitive overlap of spatial maps, specifically inanimate objects, large objects, and the place cluster from Figure 4.3 all overlap in the top left and right corners of the map. We additionally highlight the maximum activating neurons for three separate input images. We see the image of a red dresser activates a region which is simultaneously selective to places, large, and inanimate objects, echoing the nested spatial hierarchies thought by Grill-Spector and K. S. Weiner (2014) to exist in the brain. In Section A.3, we again see that such a hierarchy appears consistently across four random initalizations.

4.8. Discussion

In this chapter we have introduced a new model to the class of topographic generative models capable of integration with deep artificial neural networks. In doing so, we have demonstrated the ability topographic generative models, namely Topographic Variational Autoencoders, to model the emergence of category-selective cortical areas as well as more abstract spatial category hierarchies. We see the model agrees qualitatively with prior work and observations from neuroscience while being founded on a single information theoretic principle.

We note that this study is inherently preliminary and is limited by both the small size of the models used, as well as the feature extraction by a pre-trained convolutional model. It is possible that class-level features and even hierarchical organization are already partially present in some form in the 9216-dimensional feature vectors used as input, and thus it is unclear how much feature extraction the TVAE model is itself learning. Nevertheless, we highlight that there is nothing fundamentally limiting the TVAE framework from extending to train full deep convolutional networks end-to-end. This is in contrast to the existing related methods which require a supplementary learning signal to guide feature extraction (H. Lee et al., 2020).

In future work, we intend to explore hierarchical extensions of the TVAE, model-

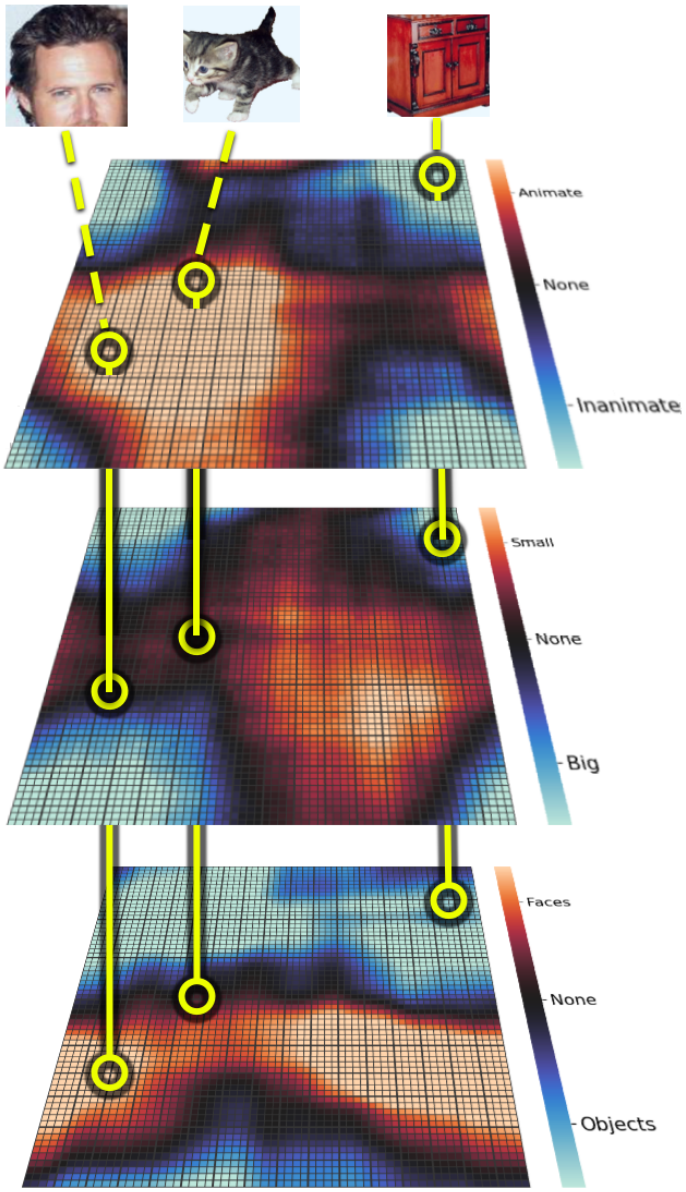


Figure 4.5: Selectivity maps for abstract categories: Animate vs. Inanimate (top), Small vs. Big (middle), and Faces vs. Objects (bottom). We highlight the maximum activating neurons for the individual images from Figure 4.4 across all maps, demonstrating their place in the proposed nested spatial hierarchy.

ing topographic organization of features at multiple levels of the visual processing pipeline while simultaneously training directly on raw pixel inputs. Such a model would validate the idea of end-to-end unsupervised category-selectivity while simultaneously providing a learned decoder from latent space to image space, opening new avenues for experimentation.

Part II

Spatio-Temporal Structure

Chapter 5

SPATIO-TEMPORAL COHERENCE INDUCES EQUIVARIANT CAPSULES

5.1. Introduction

In the previous chapter we have seen how the principle of redundancy reduction may be operationalized in order to build models which exhibit topographic organization reminiscent of that seen in natural systems. In this chapter we seek to explore an alternative link between such natural representational structure and computational theories in the machine learning community. Specifically, in this chapter we seek to investigate a second almost independant body of literature which has studied the idea of "equivariance" of neural network feature maps under symmetry transformations.

In the context of deep neural networks, the idea of equivariance is that symmetry transformations define equivalence classes as the orbits of their transformations, and we wish to maintain this structure in the deeper layers of a neural network. For instance, for images, asserting a rotated image contains the same object for all rotations, the transformation of rotation then defines an orbit where the elements of that orbit can be interpreted as pose or angular orientation. When an image is processed by a neural network, we want features at different orientations to be able to be combined to form new features, but we want to ensure the relative pose information between the features is preserved for all orientations. This has the advantage that the equivalence class of rotations for the complex composite features is guaranteed to be maintained, allowing for the extraction of invariant features, a unified pose, and increased data efficiency. Such ideas are reminiscent of the capsule networks of Hinton et al. (Geoffrey E. Hinton, Krizhevsky, et al., 2011b; Geoffrey E Hinton, Sabour, et al., 2018; Sabour et al., 2017), and indeed formal connections to equivariance have been made (Lenssen et al., 2018). Interestingly, by explicitly building neural networks to be equivariant, we additionally see geometric organization of activations into these equivalence classes, and further, the elements within an equivalence class are seen to exhibit higher-order non-Gaussian dependencies similar to those which motivated the development of topographic generative models in the previous chapter (Lyu and E. P. Simoncelli, 2008; Lyu and E. P. Simoncelli, 2009b; Wainwright and E. P. Simoncelli, 2000; Wainwright, E. P. Simoncelli, and

Willsky, 2001). The insight of this connection between topographic organization and equivariance hints at a possibility to encourage approximate equivariance from an induced topology in feature space.

To build a model which may be able to unify these ideas in a more explicit way, we need to ask what mechanisms could induce topographic organization of *observed symmetry transformations* specifically? We have shown that removing dependencies between latent variables is a possible mechanism for creating spatial structure; however, to obtain the more structured organisation of equivariant capsule representations which naturally operate over time, we can see that we will need our mechanism to be spatio-temporal rather than simply spatial. The usual approach is to hard-code this structure into the network, or to encourage it through regularization terms (Benton et al., 2020; Diaconu and D. Worrall, 2019a). To achieve this same structure *unsupervised*, we propose to incorporate another key inductive bias: "temporal coherence" (Földiák, 1991; Hurri and Aapo Hyvärinen, 2003; Stone, 1996; Wiskott and Terrence J Sejnowski, 2002). The principle of temporal coherence, or "slowness", asserts than when processing correlated sequences, we wish for our representations to change smoothly and slowly over space and time. Thinking of time sequences as symmetry transformations on the input, we desire features undergoing such transformations to be grouped into equivariant capsules. We therefore suggest that encouraging slow feature transformations to take place *within a capsule* could induce such grouping from sequences alone.

In the following sections we will explain the details of our Spatio-Temporally Coherent Variational Autoencoder which lies at the intersection of topographic organization, equivariance, and temporal coherence, thereby learning approximately equivariant capsules from sequence data completely unsupervised.

5.2. Related Work

Prior work on learning equivariant and invariant representations has a deep relationship with the generative model presented in this chapter. Specifically, Independant Subspace Analysis (Aapo Hyvärinen and P. Hoyer, 2000; Stühmer et al., 2019), models involving temporal coherence (Földiák, 1991; Hurri and Aapo Hyvärinen, 2003; Stone, 1996; Wiskott and Terrence J Sejnowski, 2002), and Adaptive Subspace Self Organizing Maps (Kohonen, 1996) have all demonstrated the ability to learn invariant feature subspaces and even 'disentangle' space and time (Grathwohl and Wilson, 2016; Stühmer et al., 2019). Our work assumes a similar generative

model to these works while leveraging the Topographic Variational Autoencoder from the previous chapter to further allow for more efficient estimation of the model through variational inference (Kingma and Max Welling, 2014; D. J. Rezende et al., 2014).

Another line of work has focused on constructing neural networks with equivariant representations separate from the framework of generative modeling. Early work in this space includes the Gaussian scale mixtures of Wainwright and E. P. Simoncelli (2000), and ensuing variants (Lyu and E. P. Simoncelli, 2009b; Portilla et al., 2003; Wainwright, E. P. Simoncelli, and Willsky, 2001). Such models were constructed specifically with the statistical correlation of transformation group elements in mind, and leveraged this correlation to define some of the first highly successful equivariant representations for image processing (E. Simoncelli and Freeman, n.d.). More recently, analytically equivariant networks such as Group Equivariant Neural Networks (T. Cohen and Max Welling, 2016a), and other extensions (T. Cohen and M. Welling, 2017; Finzi, Stanton, et al., 2020; Finzi, Max Welling, et al., 2021; Pol et al., 2020; Ravanbakhsh et al., 2017; Weiler, Geiger, et al., 2018; D. Worrall and Max Welling, 2019a; D. E. Worrall et al., 2017) propose to explicitly enforce symmetry to group transformations in neural networks through structured weight sharing. Alternatively, others propose supervised and self-supervised methods for *learning* equivariance or invariance directly from the data itself (Benton et al., 2020; Connor et al., 2021; Diaconu and D. Worrall, 2019a). One related example in this category uses a group sparsity regularization term to similarly learn topographic features for the purpose of modeling invariance (Kavukcuoglu et al., 2009). We believe the Spatio-temporal Variational Autoencoder presented in this chapter is another promising step in the direction of learning approximate equivariance, and may even hint at how such structure could be learned in biological neural networks.

Furthermore, the idea of disentangled representations (Bengio, Courville, et al., 2013) has also been been connected to equivariance and representation theory in multiple recent papers (Bouchacourt et al., 2021; Taco S. Cohen and Max Welling, 2015; T. Cohen and Max Welling, 2014; Higgins, Amos, et al., 2018). Our work shares a fundamental connection to this distributed operator definition of disentanglement, where the slow roll of capsule activations can be seen as the latent operator. Recently, the authors of (D. A. Klindt et al., 2021) demonstrated that incorporating the principle of 'slowness' in a variational autoencoder (VAE) yields the ability to learn disentangled representations from natural sequences. While similar in mo-

tivation, the generative model proposed in (D. A. Klindt et al., 2021) is unrelated to topographic organization and equivariance, and is more aligned with traditional notions of disentanglement.

5.3. The Generative Model

In this chapter, following the ideas of temporal coherence introduced by (Földiák, 1991), we will extend the Topographic VAE from the previous chapter by extending topographic neighborhoods over timesteps of an observed transformation and thereby encouraging the unsupervised learning of approximately equivariant subspaces we call 'capsules'.

Capsules as Disjoint Topologies

First let us recall the Topographic VAE formulation from Section 4.5 and specifically the construction of the topographic product of student's-t random variable from Equation 4.6: ($\mathbf{T} = \frac{\mathbf{Z}-\mu}{\sqrt{\mathbf{W}\mathbf{U}^2}}$). In this construction, the setting of the neighborhood structure $\mathbf{W}$ defines the correlations between T_i variables, effectively defining the topology of the latent variable landscape.

One setting of neighborhood structure $\mathbf{W}$ which is of particular interest to this work is when there exist multiple sets of disjoint neighborhoods. Statistically, the variables of two disjoint topologies are completely independent. An example of a capsule neighborhood structure is shown in Figure 5.1. The idea of independant subspaces has previously been shown to learn invariant feature subspaces in the linear setting and is present in early work on Independent Subspace Analysis (Aapo Hyvärinen and P. Hoyer, 2000) and Adaptive Subspace Self Organizing Maps (ASSOM) (Kohonen, 1996). It is also very reminiscent of the transformed sets of features present in a group equivariant convolutional neural network. In the next section, we will show how temporal coherence can be leveraged to induce the encoding of observed transformations into the internal dimensions of such capsules thereby yielding unsupervised approximately equivariant capsules.

Temporal Coherence and Learned Equivariance

We now describe how the induced topographic organization can be leveraged to learn a basis of approximately equivariant capsules for observed transformation sequences. The resulting representation is composed of a large set of 'capsules'

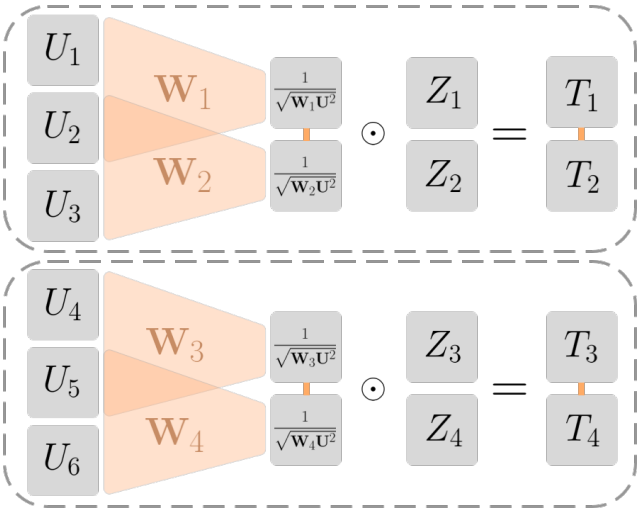


Figure 5.1: An example of a neighborhood structure which induces disjoint topologies (aka capsules). Lines between variables T_i indicate that sharing of U_i, and thus correlation.

where the dimensions inside the capsule are topographically structured, but between the capsules there is independence. To benefit from sequences of input, we encourage topographic structure over time between sequentially permuted activations within a capsule, a property we refer to as *shifting temporal coherence*.

Temporal Coherence

Temporal Coherence can be measured as the correlation of squared activation between time steps. One way we can achieve this in our model is by having T_j share U_i between time steps. Formally, the generative model is identical to Equation 4.1, factorizing over timesteps denoted by subscript l, i.e. $p_{\mathbf{X}_l,\mathbf{T}_l}(\mathbf{x}_l, \mathbf{t}_l) = p_{\mathbf{X}_l|\mathbf{T}_l}(\mathbf{x}_l|\mathbf{t}_l)p_{\mathbf{T}_l}(\mathbf{t}_l)$. However, $\mathbf{T}_l$ is now a function of a sequence $\{\mathbf{U}_{l+\delta}\}_{\delta=-L}^{L}$:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right]}} \tag{5.1}$$

Where $\left[\mathbf{U}_{l+L}^2; \cdots; \mathbf{U}_{l-L}^2\right]$ denotes vertical concatenation of the column vectors $\mathbf{U}_l$, and $2L$ can be seen as the window size. We see that the choice of $\mathbf{W}$ now defines correlation structure over time. In prior work on temporal coherence (denoted 'Bubbles' (A. Hyvärinen et al., 2004)), the grouping over time is such that a given variable $T_{l,i}$ has correlated energy with *the same spatial location* (i) at a previous

time step $(l-1)$ $\left(\text{i.e. } \text{cov}(T^2_{l,i}, T^2_{l-1,i}) > 0\right)$. This can be implemented as:

$$\mathbf{W}\left[\mathbf{U}^2_{l+L}; \cdots; \mathbf{U}^2_{l-L}\right] = \sum_{\delta=-L}^{L} \mathbf{W}_\delta \mathbf{U}^2_{l+\delta} \tag{5.2}$$

Where $\mathbf{W}_\delta$ defines the topography for a single timestep, and is typically the same for all timesteps.

Learned Equivariance with Shifting Temporal Coherence

In our model, instead of requiring a single location to have correlated energies over a sequence, we would like variables at sequentially permuted locations *within a capsule* to have correlated energy between timesteps $\left(\text{cov}(T^2_{l,i}, T^2_{l-1,i-1}) > 0\right)$. Similarly, this can be implemented as:

$$\mathbf{W}\left[\mathbf{U}^2_{l+L}; \cdots; \mathbf{U}^2_{l-L}\right] = \sum_{\delta=-L}^{L} \mathbf{W}_\delta \text{Roll}_\delta(\mathbf{U}^2_{l+\delta}) \tag{5.3}$$

Where $\text{Roll}_\delta(\mathbf{U}^2_{l+\delta})$ denotes a cyclic permutation of δ steps along the capsule dimension. The exact implementation of Roll can be found in Appendix Section B.1. As we will show in Section 5.5, TVAE models with such a topographic structure learn to encode observed sequence transformations as Rolls within the capsule dimension, analogous to a group equivariant neural network where τ_g and Roll_1 can be seen as the action of the transformation g on the input and output spaces respectively.

5.4. The Spatio-Temporally Coherent VAE

To train the parameters of the generative model, identical to Section 4.5, we build a variational autoencoder architecture capable of inferring $\mathbf{t}$ using two separate approximate posteriors for $\mathbf{u}$ and $\mathbf{z}$. Explicitly:

$$q_\phi(\mathbf{z}_l|\mathbf{x}_l) = \mathcal{N}\left(\mathbf{z}_l; \mu_\phi(\mathbf{x}_l), \sigma_\phi(\mathbf{x}_l)\mathbf{I}\right) \qquad p_\theta(\mathbf{x}_l|g_\theta(\mathbf{t}_l)) = p_\theta(\mathbf{x}_l|g_\theta(\mathbf{z}_l, \{\mathbf{u}_l\})) \tag{5.4}$$

$$q_\gamma(\mathbf{u}_l|\mathbf{x}_l) = \mathcal{N}\left(\mathbf{u}_l; \mu_\gamma(\mathbf{x}_l), \sigma_\gamma(\mathbf{x}_l)\mathbf{I}\right) \qquad \mathbf{t}_l = \frac{\mathbf{z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{u}^2_{l+L}; \cdots; \mathbf{u}^2_{l-L}\right]}} \tag{5.5}$$

To distinguish this model from the Topographic VAE in the previous chapter, we denote it the Spato-Temporally Coherent Topographic VAE (STC-TVAE) due to its reliance on a new form of (shifting) temporal coherence which moves over topographic space with respect to time. In this chapter, we will often refer to the

model simply as the TVAE given it can be seen as a generalization of the original model with an additional hyper-parameter L. We then optimize the parameters θ, ϕ, γ (and μ) through the ELBO, summed over the sequence length S:

$$\sum_{l=1}^{S} \mathbb{E}_{Q_{\phi,\gamma}(\mathbf{z}_l,\mathbf{u}_l|\{\mathbf{x}_l\})} \left([\log p_\theta(\mathbf{x}_l|g_\theta(\mathbf{t}_l))] - D_{KL}[q_\phi(\mathbf{z}_l|\mathbf{x}_l)||p_{\mathbf{Z}}(\mathbf{z}_l)] - D_{KL}[q_\gamma(\mathbf{u}_l|\mathbf{x}_l)||p_{\mathbf{U}}(\mathbf{u}_l)] \right) \tag{5.6}$$

where $Q_{\phi,\gamma}(\mathbf{z}_l, \mathbf{u}_l|\{\mathbf{x}_l\}) = q_\phi(\mathbf{z}_l|\mathbf{x}_l) \prod_{\delta=-L}^{L} q_\gamma(\mathbf{u}_{l+\delta}|\mathbf{x}_{l+\delta})$, and $\{\cdot\}$ denotes a set over time. In Figure 5.2 below, we include an overview of this model and how the Roll operation in the capsule space then corresponds to a transformation in the input space.

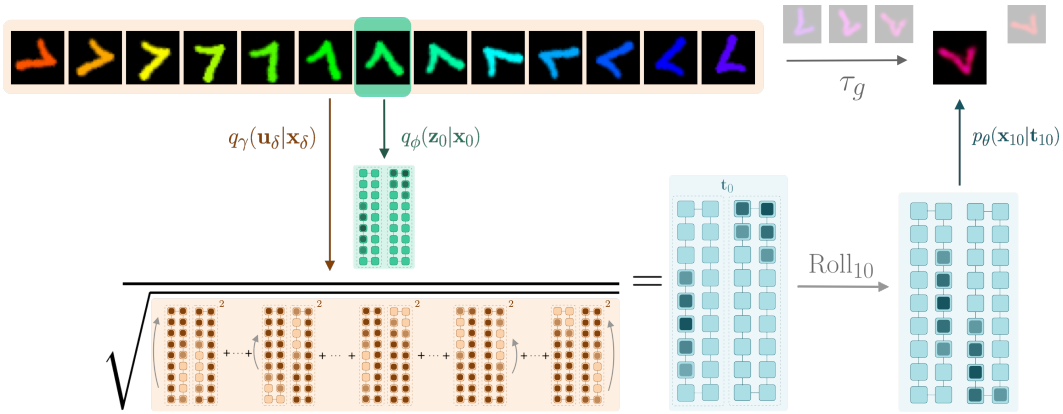


Figure 5.2: Overview of the Spatio-Temporally Coheret Topographic VAE. The combined color/rotation transformation in input space τ_g becomes encoded as a Roll within the capsule dimension. The model is thus able decode unseen sequence elements by encoding a partial sequence and Rolling activations within the capsules. We see this resembles a commutative diagram.

5.5. Experiments

In the following experiments, we demonstrate the viability of the Spatio-Temporally Coherent Topographic VAE as a novel method for learning approximately equivariant capsules by computing an 'equivariance loss' and a correlation metric inspired by the disentanglement literature. We show that equivariant capsule models yield higher likelihood than baselines on test sequences, and qualitatively support these results with visualizations of sequences reconstructed purely from Rolled capsule activations.

Evaluation Methods

As depicted in Figure 5.2, we make use of *capsule traversals* to qualitatively visualize the transformations learned by our network. Simply, these are constructed by encoding a partial sequence into a $\mathbf{t}_0$ variable, and decoding sequentially Rolled copies of this variable. Explicitly, in the top row we show the data sequence $\{\mathbf{x}_l\}_l$, and in the bottom row we show the decoded sequence: $\{g_\theta(\text{Roll}_l(\mathbf{t_0}))\}_l$.

To measure equivariance quantitatively, we measure an *equivariance error* similar to (Diaconu and D. Worrall, 2019a). The equivariance error can be seen as the difference between traversing the two distinct paths of the commutative diagram, and provides some measure of how precisely the function and the transform commute. Formally, for a sequence of length S, and $\hat{\mathbf{t}} = \mathbf{t}/||\mathbf{t}||_2$, the error is defined as:

$$\mathcal{E}_{eq}(\{\mathbf{t}_l\}_{l=1}^{S}) = \sum_{l=1}^{S-1} \sum_{\delta=1}^{S-l} \left\| \text{Roll}_\delta(\hat{\mathbf{t}}_l) - \hat{\mathbf{t}}_{l+\delta} \right\|_1 \tag{5.7}$$

Additionally, inspired by existing disentanglement metrics, we measure the degree to which observed transformations in capsule space are correlated with input transformations by introducing a new metric we call CapCorr_y. Simply, this metric computes the correlation between the amount of observed Roll of a capsule's activation at two timesteps l and $l + \delta$, and the shift of the ground truth generative factors y_l in that same time. Formally, for a correlation coefficient Corr:

$$\text{CapCorr}(\mathbf{t}_l, \mathbf{t}_{l+\delta}, y_l, y_{l+\delta}) = \text{Corr}\left(\text{argmax}\left[\mathbf{t}_l \star \mathbf{t}_{l+\delta}\right], |y_l - y_{l+\delta}|\right) \tag{5.8}$$

Where $\star$ is discrete periodic cross-correlation across the capsule dimension, and the correlation coefficient is computed across the entire dataset. We see the argmax of the cross-correlation is an estimate of the degree to which a capsule activation has shifted from time l to $l + \delta$. To extend this to multiple capsules, we can replace the argmax function with the mode of the argmax computed for all capsules. We provide additional details and extensions of this metric in Appendix Section B.1. For measuring capsule-metrics on baseline models which do not naturally have capsules, we simply arbitrarily divide the latent space into a fixed set of corresponding capsules and capsule dimensions, and provide such results as equivalent to 'random baselines' for these metrics.

Learning Equivariant Capsules

In the following experiments, we provide evidence that the Topographic VAE can be leveraged to learn equivariant capsules by incorporating shifting temporal coherence

into a 1D baseline topographic model. We compare against two baselines: standard normal VAEs and models that have non-shifting 'stationary' temporal coherence as defined in Equation 5.2 (denoted 'BubbleVAE' (A. Hyvärinen et al., 2004)).

In all experiments we use a 3-layer MLP with ReLU activations for both encoders and the decoder. We arrange the latent space into 15 circular capsules each of 15-dimensions for dSprites (Matthey et al., 2017), and 18 circular capsules each of 18-dimensions for MNIST. Example sequences $\{\mathbf{x}_l\}_{l=1}^{S}$ are formed by taking a random initial example, and sequentially transforming it according to one of the available transformations: (X-Pos, Y-Pos, Orientation, Scale) for dSprites, and (Color, Scale, Orientation) for MNIST. All transformation sequences are cyclic such that when the maximum transformation parameter is reached, the subsequent value returns to the minimum. We denote the length of a full transformation sequence by S, and the time-extent of the induced temporal coherence (i.e. the length of the input sequence) by $2L$. For simplicity, both datasets are constructed such that the sequence length S equals the capsule dimension (for dSprites this involves taking a subset of the full dataset and looping the scale 3-times for a scale-sequence). Exact details are in Appendix Sections B.1 & B.1.

In Figure 5.3, we show the capsule traversals for TVAE models with $L \approx \frac{1}{3}S$. We see that despite the $\mathbf{t}_0$ variable encoding only $\frac{2}{3}$ of the sequence, the remainder of the transformation sequence can be decoded nearly perfectly by permuting the activation through the full capsule – implying the model has learned to be approximately equivariant to full sequences while only observing partial sequences per training point. Furthermore, we see that the model is able to successfully learn all transformations simultaneously for the respective datasets.

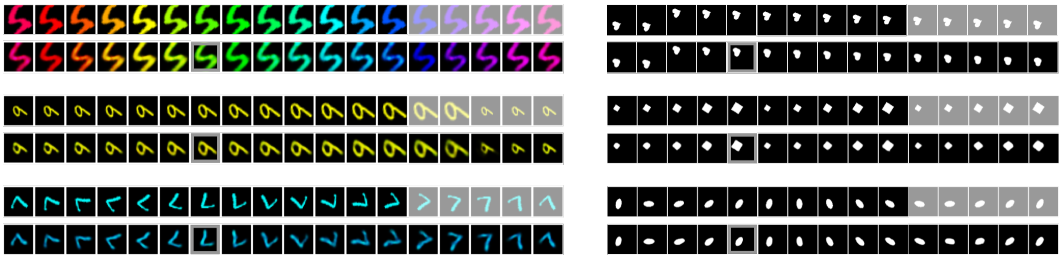


Figure 5.3: Capsule Traversals for TVAE models on dSprites and MNIST. The top rows show the encoded sequences (with greyed-out images held-out), and the bottom rows show the images generated by decoding sequentially Rolled copies of the initial activation $\mathbf{t}_0$ (indicated by a grey border).

Capsule traversals for the non-equivariant baselines, as well as TVAEs with smaller

values of L (which only learn approximate equivariance to partial sequences) are shown in Appendix Section B.4. We note that the capsule traversal plotted in Figure 5.2 demonstrates a transformation where color and rotation change simultaneously, differing from how the models in this section are trained. However, as we describe in more detail in Section B.2, we observe that TVAEs trained with individual transformations in isolation (as in this section) are able to generalize, generating sequences of combined transformations when presented with such partial input sequences at test time. We believe this generalization capability to be promising for data efficiency, but leave further exploration to future work. Additional capsule traversals with such unseen combined transformations are shown in Section B.2 and further complex learned transformations (such as perspective transforms) are shown at the end of Section B.4.

For a more quantitative evaluation, in Table 5.1 we measure the equivariance error and log-likelihood (reported in nats) of the test data under our trained MNIST models as estimated by importance sampling with 10 samples. We observe that models which incorporate temporal coherence (BubbleVAE and TVAE with $L > 0$) achieve low equivariance error, while the TVAE models with shifting temporal coherence achieve the highest likelihood and the lowest equivariance error simultaneously.

Table 5.1: Log Likelihood and Equivariance Error on MNIST for different settings of temporal coherence length L relative to sequence length S. Mean ± std. over 3 random initalizations.

Model	TVAE	TVAE	TVAE	BubbleVAE	VAE
L	$L = \frac{1}{2}S$	$L = \frac{5}{36}S$	$L = 0$	$L = \frac{5}{36}S$	$L = 0$
$\log p(\mathbf{x}) \uparrow$	$\mathbf{-186.8} \pm 0.1$	$\mathbf{-186.0} \pm 0.7$	-218.5± 0.9	-191.4 ± 0.5	-189.0 ± 0.8
$\mathcal{E}_{eq} \downarrow$	$\mathbf{574} \pm 2$	3247 ± 3	3217 ± 105	3370 ± 12	13274 ± 1

To further understand how capsules transform for observed input transformations, in Table 5.2 we measure $\mathcal{E}_{eq}$ and the CapCorr metric on the dSprites dataset for the four proposed transformations. We see that the TVAE with $L \geq \frac{1}{3}S$ achieves perfect correlation – implying the learned representation indeed permutes cyclically within capsules for observed transformation sequences. Further, this correlation gradually decreases as L decreases, eventually reaching the same level as the baselines. We also see that, on both datasets, the equivariance losses for the TVAE with $L = 0$ and the BubbleVAE are significantly lower than the baseline VAE, while conversely, the CapCorr metric is not significantly better. We believe this to be due to the

fundamental difference between the metrics: $\mathcal{E}_{eq}$ measures continuous L1 similarity which is still low when a representation is locally smooth (even if the change of the representation does not follow the observed transformation), whereas CapCorr more strictly measures the correspondence between the transformation of the input and the transformation of the representation. In other words, $\mathcal{E}_{eq}$ may be misleadingly low for invariant capsule representations (as with the BubbleVAE), whereas CapCorr strictly measures equivariance.

Table 5.2: Equivariance error ($\mathcal{E}_{eq} \downarrow$) and correlation of observed capsule roll with ground truth factor shift (CapCorr $\uparrow$) for the dSprites dataset. Mean ± standard deviation over 3 random initalizations.

Model	TVAE	TVAE	TVAE	TVAE	BubbleVAE	VAE
L	$L = \frac{1}{2}S$	$L = \frac{1}{3}S$	$L = \frac{1}{6}S$	$L = 0$	$L = \frac{1}{3}S$	$L = 0$
$\text{CapCorr}_X \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.67 ± 0.02	0.17 ± 0.03	0.13 ± 0.01	0.18 ± 0.01
$\text{CapCorr}_Y \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.66 ± 0.02	0.21 ± 0.02	0.12 ± 0.01	0.16 ± 0.01
$\text{CapCorr}_O \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.52 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.11 ± 0.00
$\text{CapCorr}_S \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.42 ± 0.01	0.51 ± 0.01	0.50 ± 0.00	0.52 ± 0.00
$\mathcal{E}_{eq} \downarrow$	$\mathbf{344} \pm 5$	1034 ± 6	2549 ± 38	2971 ± 9	1951 ± 34	6934 ± 0

5.6. Building a Forward Predictive Model

From this analysis, there are two clear extensions that can be made to make the model more amenable to traditional sequence modeling while additionally including additional priors from the neuroscience community.

First, considering the inputs $\{\boldsymbol{x}_l\}_{l=0}^{S}$ to our model as a time-sequence, it would be beneficial if the model did not 'look into the future' when reconstructing the current time step. A visualization of this can be seen in Figure 5.2, where the model is aiming to reconstruct the green '7' digit in the center, but it uses the blue digits 'in the future' in order to encode the denominator of the **T** variable. We would therefore like to impose a constraint on the model to only consider 'past' observations when reconstructing the present. In the computer vision and natural language processing literature, this constraint on a convolution operator is called making the convolution 'causal', and it is mainly used when a convolutional model is being used in sequence modeling. In our model, since we use convolutions in practice in order to define the neighborhood structure $\boldsymbol{W}$, it is therefore relatively easy to implement such a constraint.

Secondly, one well known generative modeling framework in the theoretical neuroscience literature is that of predictive coding (Elias, 1955; Rao and Ballard, 1999; K. Friston, 2005; Clark, 2013). At a high level, predictive coding denotes a framework by which the cortex is a generative model of sensory inputs, and has been linked to probabilistic latent variable models such as VAEs (Marino, 2020). Substantial evidence has been gathered supporting the existence of some form of predictive coding in the brain (Alink et al., 2010; Ouden et al., 2010; Egner et al., 2010), and numerous computational models have been proposed which replicate empirical observations (Rao and Ballard, 1999; Lotter et al., 2018; G. B. Keller and Mrsic-Flogel, 2018). Given these computational successes, and the mounting support for such a mechanism underlying biological intelligence, we seek to understand if there may be a relationship between predictive coding and Topographic VAEs.

In the context of the Topographic VAE, since our 'sensory input' is entirely visual, we can define our prediction goal as a simple forward predictive model of future observations. Given that in the previous sections we have already introduced a model with spato-temporally organized capsules, we see that our model will therefore permit efficient forward prediction through simple forward rolling of capsule activations. Leveraging such forward prediction to create a generative model of the immediate future permits online learning and inference in the TVAE, increasing flexibility of the original model. Furthermore, as we demonstrate empirically in this section, such a model is able to more accurately predict the immediate future, while simultaneously retaining the learned equivariance properties afforded by the original Spatio-Temporally Coherent TVAE. At a high level, our goal is thus to modify the model in Figure 5.2 to look more like the model in Figure 5.4, which we denote the Predictive Coding Topographic VAE.

In detail, we accomplish this through two changes to the STC-TVAE from earlier in this section – we make the conditional generative distribution forward predictive, and limit the temporal coherence window to only include past variables.

Past Temporal Coherence

As mentioned in the Section 5.3, the Spatio-Temporally Coherent Topographic VAE takes advantage of the generalized framework of topographic generative models to induce structured correlations of activations over time – thereby achieving equivariance. To limit these correlations to only include past variables, we define $\mathbf{T}_l$ as a function of a sequence $\{\mathbf{U}_{l-\delta}\}_{\delta=0}^{L}$, defining $\mathbf{W}$ to connect sequentially rolled copies

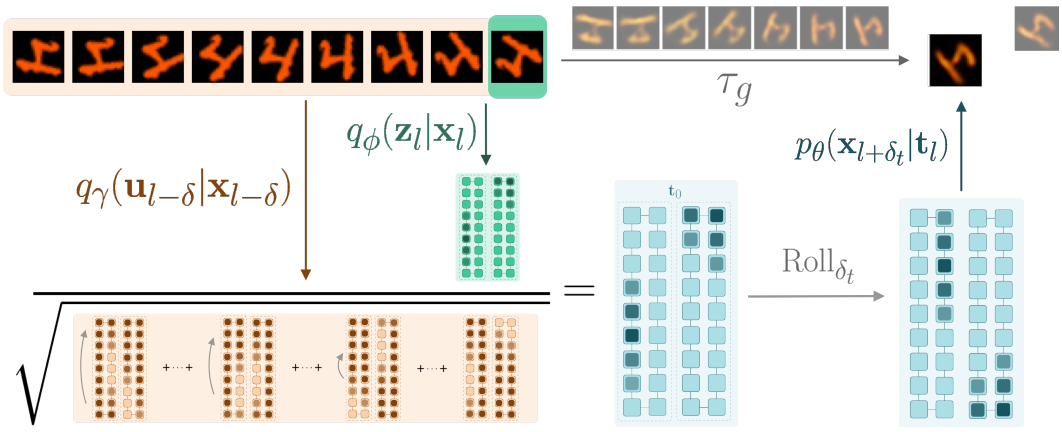


Figure 5.4: Overview of the Predictive Coding Topographic VAE. The transformation in input space τ_g becomes encoded as a Roll within the equivariant capsule dimension. The model is thus able to forward predict the continuation of the sequence by encoding a partial sequence and rolling activations within the capsules.

of past $\mathbf{U}_l$:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{U}_l^2; \cdots; \mathbf{U}_{l-L}^2\right]}} \tag{5.9}$$

where $\left[\mathbf{U}_l^2; \cdots; \mathbf{U}_{l-L}^2\right]$ denotes vertical concatenation of the column vectors $\mathbf{U}_l$, and L can be seen as the past window size. Then, by careful definition of $\mathbf{W}$, we can achieve the same 'shifting temporal coherence', defined above, yielding equivariant capsules. Explicitly, $\mathbf{W}$ is thus given by:

$$\mathbf{W}\left[\mathbf{U}_l^2; \cdots; \mathbf{U}_{l-L}^2\right] = \sum_{\delta=0}^{L} \mathbf{W}_\delta \mathrm{Roll}_\delta(\mathbf{U}_{l-\delta}^2) \tag{5.10}$$

where $\mathbf{W}_\delta$ defines a set of disjoint 'capsule' topologies for each time-step, and $\mathrm{Roll}_\delta(\mathbf{U}_{l-\delta}^2)$ denotes a cyclic permutation of δ steps along the capsule dimension.

The Predictive Coding TVAE

Finally, as before, we use 5.9 to parameterize an approximate posterior for $\mathbf{t}_l$ in terms of a deterministic transformation of approximate posteriors over simpler Gaussian latent variables $\mathbf{z}_l$ and $\mathbf{u}_l$:

$$q_\phi(\mathbf{z}_l|\mathbf{x}_l) = \mathcal{N}\left(\mathbf{z}_l; \mu_\phi(\mathbf{x}_l), \sigma_\phi(\mathbf{x}_l)\mathbf{I}\right) \tag{5.11}$$

$$q_\gamma(\mathbf{u}_l|\mathbf{x}_l) = \mathcal{N}\left(\mathbf{u}_l; \mu_\gamma(\mathbf{x}_l), \sigma_\gamma(\mathbf{x}_l)\mathbf{I}\right) \tag{5.12}$$

$$\mathbf{t}_l = \frac{\mathbf{z}_l - \mu}{\sqrt{\mathbf{W}\left[\mathbf{u}_l^2; \cdots; \mathbf{u}_{l-L}^2\right]}} \tag{5.13}$$

Additionally, to further encourage the capsule Roll as the forward prediction operator, we integrate a capsule Roll of $\mathbf{t}_l$ by one unit as the first step of the generative model, before decoding $\mathbf{x}_{l+1}$:

$$p_\theta(\mathbf{x}_{l+1}|g_\theta(\mathbf{t}_l)) = p_\theta(\mathbf{x}_{l+1}|\hat{g}_\theta(\text{Roll}_1[\mathbf{t}_l])) \tag{5.14}$$

We denote this model the Predictive Coding Topographic VAE (PCTVAE) and present an overview of forward prediction in Figure 5.2. We optimize the parameters θ, ϕ, γ (and μ) through the ELBO, summed over the sequence length S:

$$\begin{aligned}\sum_{l=1}^{S} \mathbb{E}_{Q_{\phi,\gamma}(\mathbf{z}_l,\mathbf{u}_l|\{\mathbf{x}\})}\Big(&\log p_\theta(\mathbf{x}_{l+1}|\hat{g}_\theta(\text{Roll}_1[\mathbf{t}_l])) \\ &- D_{KL}[q_\phi(\mathbf{z}_l|\mathbf{x}_l)||p_\mathbf{Z}(\mathbf{z}_l)] \\ &- D_{KL}[q_\gamma(\mathbf{u}_l|\mathbf{x}_l)||p_\mathbf{U}(\mathbf{u}_l)]\Big)\end{aligned} \tag{5.15}$$

where $Q_{\phi,\gamma}(\mathbf{z}_l, \mathbf{u}_l|\{\mathbf{x}\}) = q_\phi(\mathbf{z}_l|\mathbf{x}_l) \prod_{\delta=0}^{L} q_\gamma(\mathbf{u}_{l-\delta}|\mathbf{x}_{l-\delta})$. The fundamental differences of this model with the TVAE are that this model is trained to maximize the likelihood of *future* inputs through the Roll operation present in the ELBO, and that the construction of $\mathbf{t}_l$ is now only a function of past inputs. As we will demonstrate in the next subsection, these extensions yield significant improvements to sequence modeling, while simultaneously increasing flexibility by allowing for online training and inference.

In the following subsections we measure the performance of our model, compared with the original TVAE and a standard VAE baselines, on the transforming color MNIST dataset from the previous section. We additionally use the same model architectures as before for valid comparison.

Forward Prediction Likelihood

To quantitatively measure the ability of the PCTVAE to predictively model sequences, we train the model to maximize Equation 5.15 with stochastic gradient descent, and measure the likelihood of held-out test sequences, with only partial sequences as input. Explicitly, for both the (STC)-TVAE and PCTVAE, a window size of 9 observations are provided as input and used to generate a capsule representation $\mathbf{t}_0$. The likelihood of the remaining 9 sequence elements is then measured by sequentially rolling the capsule activations forward, and measuring $p_\theta(\mathbf{x}_{\delta_t}|g_\theta(\text{Roll}_{\delta_t}(\mathbf{t_0})))$ for $\delta_t \in \{0, ..., 9\}$. The final reported likelihood values are

	NLL @ $\delta_t = 0$	NLL Avg. Seq.	$\mathcal{E}_{eq}$ Avg. Seq.
VAE	190 ± 1	N/A	13274 ± 0
TVAE	**187** ± 1	452 ± 16	2122 ± 21
PCTVAE	207 ± 1	**232** ± 1	2201 ± 9

Table 5.3: Neg. log-likelihood (NLL in nats) without forward prediction ($\delta_t = 0$), NLL averaged over the forward predicted sequence, and equivariance error $\mathcal{E}_{eq}$ for a non-topographic VAE, TVAE, and PCTVAE. The PCTVAE achieves the lowest average NLL over the forward predicted sequence while also maintaining low equivariance error. Mean ± std. over 3 random initalizations.

computed by importance sampling with 10 samples. In Table 5.3 we report the average log-likelihood over this forward predicted sequence for both the TVAE of Section 5.4 and the PCTVAE, in addition to the log-likelihood at $\delta_t = 0$ (no forward prediction) with a standard VAE. We see the PCTVAE achieves a significantly lower average negative likelihood in the forward prediction task, while maintaining a similar level of approximate equivariance as measured by the equivariance error $\mathcal{E}_{eq}$. We omit the baseline VAE for the sequence likelihood measurements since it has no defined forward prediction operation.

In Figure 5.5, we plot the likelihood of future sequence elemets as a function of the forward time offset δ_t. As can be seen, the TVAE model has a marginally higher likelihood for $\delta_t = 0$, but its forward predictive performance rapidly deteriorates as the capsule is rolled forward. Conversely, the PCTVAE is observed to obtain consistently high likelihoods on forward prediction up to 8 steps into the future of the sequence, implying it has learned to capture the transformation sequence structure more accurately. Interestingly, despite the TVAE actually being provided with an input window extending to $\delta_t \leq 4$ (as seen in Figure 5.6 right), the PCTVAE yields significantly higher likelihoods even for these immediate-future observations.

Sequence Generation

As a qualitative evaluation of the PCTVAE's sequence modeling capacity, we show forward predicted sequences generated by both models in Figure 5.6. The top row shows the input sequence with grey images held out, and the lower row shows the forward predicted sequence, generated by sequentially rolling the representation $\mathbf{t}_0$ forward, and decoding at each step. As can be seen, the PCTVAE (left) appears to generate sequences which are more coherent with the provided input sequence,

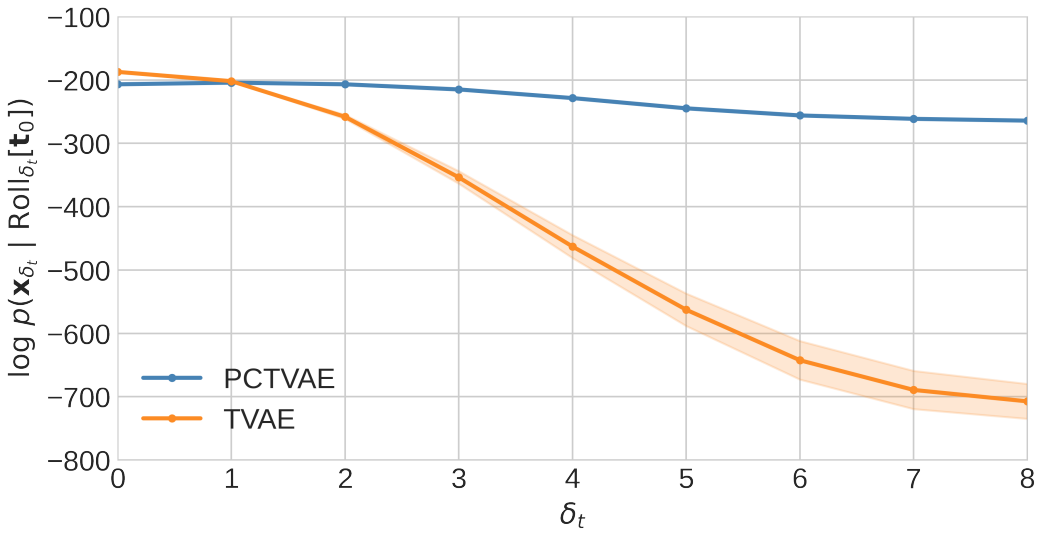


Figure 5.5: Forward prediction log-likelihood vs. future time offset δ_t. We see that the PCTVAE has consistently high likelihood for sequence elements into the future whereas the likelihood of the TVAE model drops off rapidly. Shading denotes ± 1 std.

while the TVAE (right) is observed to quickly diverge from the true transformation, in agreement with likelihood values of Figure 5.5.

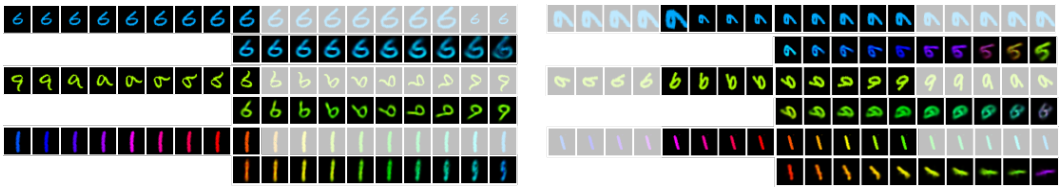


Figure 5.6: Forward predicted trajectories from the Predictive Coding TVAE (left) and the original TVAE (right). The images in the top row show the true input transformation, with greyed out images being held out. The lower row then shows the reconstruction, constructed by starting at $\mathbf{t}_0$, and progressively rolling the capsules forward to decode the remainder of the sequence. We see the PCTVAE is able to predict sequence transformations accurately, while the TVAE forward predictions slowly lose coherence with the input sequence.

5.7. Future Work & Limitations

In future work, it would be valuable to explore the ability of the models introduced in this section to learn more realistic transformations from natural data, such as from the Natural Sprites dataset (D. A. Klindt et al., 2021). Furthermore, it would be interesting to further investigate the downstream computational benefits gained from the learned equivariant capsule representation. Specifically, further study of the TVAE

both with and without temporal coherence in terms of the sample complexity, semi-supervised classification accuracy, and invariance through structured topographic pooling would be enlightening as to the additional computational benefits of this approach.

Despite the successes of the proposed models, they do admit a number of limitations in their existing form which we believe to be interesting directions for future research. First, some model developers may find the a priori definition of topographic structure burdensome. While true, we know that the construction of appropriate priors is always a challenging task in latent variable models, and we observe that our proposed TVAE achieves strong performance even with improper specification. Furthermore, in future work, we believe adding learned flexibility to the parameters $\mathbf{W}$ may alleviate some of this burden.

As more significant limitations, the model is challenging to compare directly with existing disentanglement and equivariance literature since it requires an input sequence which determines the transformations reachable through the capsule roll. By transitioning the model to a true recurrent neural network, this comparison and integration into existing deep neural networks would be much easier.

Finally, we note that the latent operator which induces the approximate equivariance of our model (Roll) must be fixed a priori by the model developers. While the shape of the capsules and the details of the roll operation can be tuned, the format of the operation is relatively rigid. For a model to learn approximate equivariance with respect to a much broader range of transformations, we would like this latent operator to also be learned simultaneously with the operation of the model.

In the following chapters, we will present models which exactly address these final two limitations, yielding a recurrent neural network which has a more flexibly learned form of approximate equivariance.

5.8. Conclusion

In the above work we introduce the Spatio-Temporally Coherent Topographic Variational Autoencoder and show how topography can be leveraged to learn approximately equivariant sets of features, a.k.a. capsules, directly from sequences of data with no other supervision. Ultimately, we believe these results may shine some light on how biological systems could hard-wire themselves to more effectively learn representations with equivariant capsule structure.

Chapter 6

TRAVELING WAVES AS GENERALIZED SPATIO-TEMPORAL COHERENCE

6.1. Introduction

In the previous chapter, we have seen how the mechanism underlying topographic organization in our model may be extended to the time dimension, thereby yielding spatio-temporal organization and ultimately approximately equivariant capsules. In doing so, we were required to specify the precise space-time neighborhood structure for our topographic latent variable model which then determined the resulting flow of activity in latent space for a given input transformation. We likened this latent flow of activity to the latent operator in equivariant neural networks, thereby demonstrating that our model had achieved a form of approximate equivariance to observed transformations. Although this alleviated the need for the model designer to a priori specify the desired transformation groups to which the model should be equivariant with respect to, and the model was instead able to learn those directly from sequences of data, the burden of design then instead shifted to this new latent operator and capsule structure. In this chapter, we consider if there might be alternative natural forms of representational structure which could serve to act as this latent operator, while maintaining the ability to be flexibly learned while modeling the data.

As noted in the background Section 2.2, one such form of structure which has recently gained increasing interest in the neuroscience community is that of traveling waves of neural activity. Such waves have been measured at both local (Davis, Muller, et al., 2020) and global (Muller, Piantoni, et al., 2016) scales, and have been shown to be strongly related to alpha, theta, and gamma oscillations in a variety of brain regions (Honghui Zhang et al., 2018; Besserve et al., 2015). Prompted by these observations, a large number of theoretical hypotheses have been developed which attempt to explain the computational purposes of traveling waves (Muller, Chavane, et al., 2018), and the inductive biases which they may mediate.

Of particular relevance to the machine learning community, one hypothesis is that traveling waves serve to beneficially structure neural representations in both space and time (Lubenov and Siapas, 2009; Jancke et al., 2004), acting as an inductive

bias towards similarly structured natural data. As we have described throughout this thesis, structured representations have been previously demonstrated in the machine learning community to be extremely valuable, making learning models both more efficient and robust (T. Cohen and Max Welling, 2016a; D. E. Worrall et al., 2017). It is thus suggested that traveling waves may facilitate a similar kind of spatiotemporal structure in neural representations, thereby granting the observed robustness and efficiency of natural intelligence which is still lacking in modern deep neural networks (Lake et al., 2017). To date, however, testing ideas related to the computational purposes of traveling waves has been challenging due to a lack of neural network architectures which have a notion of spatial locality necessary for modeling such spatio-temporal dynamics. Further, existing networks which do have such spatial structure often do not have temporal structure (H. Lee et al., 2020), or are not sufficiently flexibly parameterized to allow them to be trained on standard machine learning benchmarks (Davis, G. B. Benigno, et al., 2021).

In this chapter, we propose to investigate the computational hypotheses surrounding traveling waves through a bottom-up approach; we build a flexibly parameterized computational model known to be capable of producing traveling waves, and show that it indeed learns to exhibit complex spatiotemporal dynamics when modeling real data. We then show, relevant to the computational neuroscience community, how such a network indeed learns spatial and temporal structure reminiscent of that found in the brain. Specifically, we observe that our network learns topographically organized selectivity, similar to the observed orientation columns and hypercolumns of the primary visual cortex (Torsten N. Wiesel and David H. Hubel, 1974). Further, we show that our network learns to use complex spatiotemporal organization such as traveling waves to encode transformations by artificially inducing waves in the hidden state and observing that this allows us to further progress or reverse the transformations of generated images.

As it relates to inductive biases, we asses the computational implications of the observed representational structure by training the model on the physical dynamics forecasting suite introduced in the paper 'Which Priors Matter?' (Botev et al., 2021). We see that our model is more accurate at predicting future trajectories of simple physical dynamics when compared with existing state of the art models, providing evidence that the structure mediated by traveling waves is indeed a beneficial inductive bias for modeling such smooth natural transformations. Further, due to our model's local connectivity, we see that it is more efficient both in terms of

parameters, and in terms of biological concerns such as wiring length, suggesting a connection between locality of connections, waves, and an inductive transformation bias in biological systems.

Overall in this chapter we introduce new powerful model at the interface of computational neuroscience and modern machine learning. We show that this model allows for the investigation of the computational hypotheses surrounding complex synchrony in the brain in a new way, and further provides preliminary evidence for the existing hypothesis that traveling waves serve to induce spatiotemporal structure in neural representations.

6.2. Background

Traveling Waves in Neuroscience

Neural oscillations and traveling waves have long been a subject of study in neuroscience and neurophysiology (Hughes, 1995; Muller, Chavane, et al., 2018). Although such waves were originally measured primarily in anesthetized subjects, improved multi-channel recording and analysis techniques have recently demonstrated propagating wave activity in awake functioning subjects as well, originating from both external stimuli and internal 'spontaneous' recurrent connections (T. K. Sato et al., 2012; Muller, Reynaud, et al., 2014; Muller, Chavane, et al., 2018). While many hypotheses have been put forth for their precise computational role, a consensus has yet to be reached. Example hypotheses include that traveling waves may: influence visual perception (Zanos et al., 2015); modulate information transfer (Besserve et al., 2015); correlate with conscious awareness (Bhattacharya, Donoghue, et al., 2022); facilitate predictive coding (K. J. Friston, 2019; Alamia and VanRullen, 2019); lower the threshold for detection of weak stimuli (Davis, Muller, et al., 2020); serve as a short term memory (King and Wyart, 2021; Bhattacharya, Brincat, et al., 2022); or as a mechanism for the formation of long-term memories during sleep (Muller, Piantoni, et al., 2016). Relevant to this work, traveling waves have directly been implicated in the encoding of motion (Heitmann and G. Bard Ermentrout, 2020), and have been measured to correlate strongly with perceived perceptual illusions of motion (Jancke et al., 2004). Further, it has been suggested that they form the basis of alpha and theta oscillations (Honghui Zhang et al., 2018; Lubenov and Siapas, 2009) and may serve to both structure and integrate information across space and time (T. K. Sato et al., 2012; N. Sato, 2022). Due to the fundamental relationship between neural synchrony and the coordination of spike

timing (Bragin et al., 1995), it is natural to wonder if more complex forms of spatiotemporal synchrony such as traveling waves may play a similarly more complex structural role.

Computational Models of Traveling Waves

In the fields of computational and theoretical neuroscience, multiple models have been developed to help explain the observed complex synchronous dynamics of neural systems. One classical model is that of a network of locally coupled oscillators (Diamant and Bortoff, 1969; George Bard Ermentrout and Kopell, 1984). However, to date, such models have been limited to those which either are built for the primary purpose of analysis (Kuramoto, 1981; G Bard Ermentrout and Kleinfeld, 2001; Davis, G. B. Benigno, et al., 2021), or those which perform very simple binary operations (Gong and Leeuwen, 2009; Izhikevich and Hoppensteadt, 2008), with neither set leveraging the flexible computational capabilities of modern deep neural networks. One line of work has aimed to integrate classical Kuramoto models into deep neural networks by directly parameterizing activations in terms of phase values (Ricci et al., 2021), however such models lack a notion of spatial locality, making the existence of spatio-temporal dynamics less concrete. Most recently, Davis, G. B. Benigno, et al. (2021) studied a large scale locally connected spiking neural network model, quantifying the conditions necessary for the emergence of traveling waves, and showed such waves appeared to uniquely agree with human cortical traveling waves in a variety of dimensions. However, similar to most existing models in this category, the model is formulated as a spiking neural network thus requiring more sophisticated training mechanisms which are yet to scale to the same performance as deep neural networks (Neftci et al., 2019).

6.3. Neural Wave Machines

In the following section we introduce the Neural Wave Machine (NWM), a deep neural network architecture which exhibits traveling waves and other complex spatiotemporal dynamics in the service of flexible differentiable computation. To achieve this, we take inspiration from the seminal models of traveling waves built as networks of locally coupled oscillators (G Bard Ermentrout and Kleinfeld, 2001), and propose to integrate them into a modern deep learning framework by taking advantage of the recently developed coupled oscillatory Recurrent Neural Network (coRNN) of T. Konstantin Rusch and Mishra (2021a).

Coupled Oscillatory Recurrent Neural Networks

In (T. Konstantin Rusch and Mishra, 2021a) the authors propose to solve the Exploding and Vanishing Gradient Problem (EVGP) in recurrent neural networks by defining a new recurrent neural network with hidden state dynamics given by the parameterized equations of a system of coupled, damped, and driven oscillators. Explicitly, the hidden state of the recurrent neural network $\mathbf{x}$ is updated by solving the following second order partial differential equation:

$$\ddot{\mathbf{x}} = \sigma\left(\mathbf{W}_x\mathbf{x} + \mathbf{W}_{\dot{x}}\dot{\mathbf{x}} + \mathbf{V}\mathbf{u} + \mathbf{b}\right) - \gamma\mathbf{x} - \alpha\dot{\mathbf{x}} \tag{6.1}$$

Where $\frac{\partial \mathbf{x}}{\partial t} = \dot{\mathbf{x}}$, $\frac{\partial^2 \mathbf{x}}{\partial t^2} = \ddot{\mathbf{x}}$ are the first and second derivatives of the hidden state with respect to time, and $\mathbf{u}$ denotes the input at each time step. The terms $\mathbf{W}_x\mathbf{x}$, $\mathbf{W}_{\dot{x}}\dot{\mathbf{x}}$, and $\mathbf{V}\mathbf{u}$ can then be interpreted as the coupling, damping, and driving terms respectively. Finally, σ is a nonlinear activation function such as the hyperbolic tangent, and γ & α are scalar variables which can be fixed or learned in combination with the above matrices. In practice, the above differential equation can be discretized and integrated numerically using an IMEX (implicit-explicit) discretization scheme shown to preserve the desirable bounds of the continuous system. Such a discretization can be achieved by first introducing a 'velocity' variable $\mathbf{v} = \dot{\mathbf{x}}$, turning the second order system into a set of two coupled first order equations:

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \sigma\left(\mathbf{W}_x\mathbf{x} + \mathbf{W}_{\dot{x}}\mathbf{v} + \mathbf{V}\mathbf{u} + \mathbf{b}\right) - \gamma\mathbf{x} - \alpha\mathbf{v} \tag{6.2}$$

Then, for a fixed time step $0 < \Delta t < 1$, the hidden state $\mathbf{x}$ and velocity $\mathbf{v}$ of the RNN at time $t + 1$ can be updated as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t(\mathbf{v}_{t+1}) \qquad \mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t(\mathbf{v}'_t) \tag{6.3}$$

$$\mathbf{v}'_t = \sigma\big(\mathbf{W}_x\mathbf{x}_t + \mathbf{W}_{\dot{x}}\mathbf{v}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}\big) - \gamma\mathbf{x}_t - \alpha\mathbf{v}_t \tag{6.4}$$

This model was theoretically demonstrated to have a bounded gradient and hidden state magnitude under assumptions on the time-step Δt and the infinity norm of the coupling parameters. Empirically, such stable gradient dynamics were shown to yield better performance than existing RNNs on tasks with very long time-dependencies.

In relation to our goals, the oscillatory dynamics of the coRNN make it amenable to synchronous activity, unlike most existing deep neural network models, and the stable gradient dynamics make it a powerful and flexibly parameterizable sequence model, unlike existing models of traveling waves based on spiking neural networks.

However, given that the hidden state $\mathbf{x}$ is not endowed with any notion of spatial layout, it is still not meaningful to study spatiotemporal dynamics in such a model. In the following subsection we describe how such a spatial layout may be implemented efficiently by replacing the fully connected recurrent coupling matrices $\mathbf{W}_x$ and $\mathbf{W}_{\dot{x}}$ with convolution operations.

Local Connectivity

In (Davis, G. B. Benigno, et al., 2021), the authors study a large scale spiking neural network model, quantifying the emergence of traveling waves, and comparing them with waves observed in the human cortex. At a high level, as it is relevant to this work, the study concludes that locally restricted connectivity and distance dependant conduction delays are both necessary and sufficient to produce traveling waves. Further they observe that such waves are fairly robust to the synaptic strengths of their model when given a sufficiently large number of neurons. Given these findings, we hypothesize that the Coupled Oscillitory Recurrent Neural Network may yield traveling waves if similarly constrained.

To impose such constraints we begin by defining an arbitrary topographic layout for the N-dimensional hidden state $\mathbf{x}$ in the model. For computational simplicity, we propose to use a regular 1 or 2 dimensional grid, $\mathbf{x}_{1D} \in \mathbb{R}^{C_h \times N}$ or $\mathbf{x}_{2D} \in \mathbb{R}^{C_h \times \sqrt{N} \times \sqrt{N}}$ respectively, where C_h is the number of simultaneous 'channels' in our hidden state. We then see that specifically, if the recurrent connections $\mathbf{W}_x$ and $\mathbf{W}_{\dot{x}}$ are made local over our spatial dimensions rather than global, and a distance-dependant time-delay introduced, the aforementioned constraints will be satisfied and the remainder of the properties such as synaptic strength and the precise local distribution of connections will be left up to the model to learn. In practice, we simplify the model by restricting the topographic connectivity of each neuron to its immediately adjacent neighbors in the grid, and define all distances (and thus time-delays) to these neurons to be equal to 1. Such a simplification allows us to efficiently implement the local time-delayed connections with a simple size 3 or 3×3 convolutional kernel for 1 and 2 dimensional grids respectively. In summary, our model is then given identically as in Equations 6.3 & 6.4 but with convolutional layers in place of the dense recurrent matrices. Explicitly, in the 2-dimensional setting, for convolutional kernels $\mathbf{w}_x, \mathbf{w}_{\dot{x}} \in \mathbb{R}^{C_h \times C_h \times 3 \times 3}$, we get:

$$\mathbf{v}'_t = \sigma\left(\mathbf{w}_x \star \mathbf{x}_t + \mathbf{w}_{\dot{x}} \star \mathbf{v}_t + f_\theta(\mathbf{u}_{t+1}) + \mathbf{b}\right) - \gamma \mathbf{x}_t - \alpha \mathbf{v}_t \tag{6.5}$$

We see we have additionally replaced the linear encoder $\mathbf{V}$ with a function f_θ which can be a convolutional or 'de-convolutional' neural network, or any other mapping from the input to a spatially organized driving force. Importantly, we see that our imposed local connectivity does not immediately invalidate any of the assumptions required for the theorems of T. Konstantin Rusch and Mishra (2021a) about mitigating the EVGP since the infinity norm of the weights is unlikely to significantly increase when simply switching from fully to locally-connected matrices. We include the updated bounds and corresponding proofs in Appendix C.2. In the end, we denote this model the Neural Wave Machine due to its emergent wave-like dynamics, facilitated by both the oscillatory update equations of the coRNN, and the local connectivity constraints of biological models. In the next section we measure these desired spatiotemporal dynamics of the NWM and further study their impact as an inductive bias on computation.

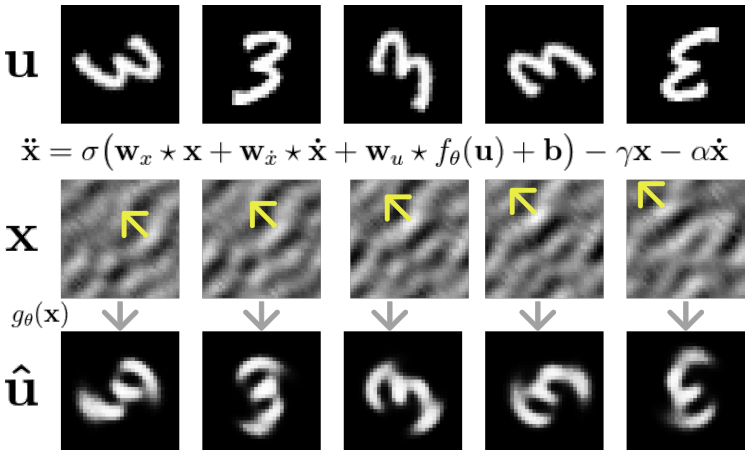


Figure 6.1: Overview of the Neural Wave Machine. The input sequence $\mathbf{u}$ is encoded with f_θ to act as a driving term in the hidden state $\mathbf{x}$ which is modeled temporally ($\mathbf{x}$) as a network of locally coupled oscillators. The network is then trained to reconstruct the input sequence: $\hat{\mathbf{u}} = g_\theta(\mathbf{x})$. The yellow arrows track a traveling wavefront over time.

6.4. Experiments

In the following two subsections we provide experiments which demonstrate: first, that our model learns spatiotemporal structure reminiscent of natural observations from neuroscience; and second, that such structure is beneficial to both efficiency and accuracy. We outline our methods briefly below, and more thoroughly in Appendix C.1.

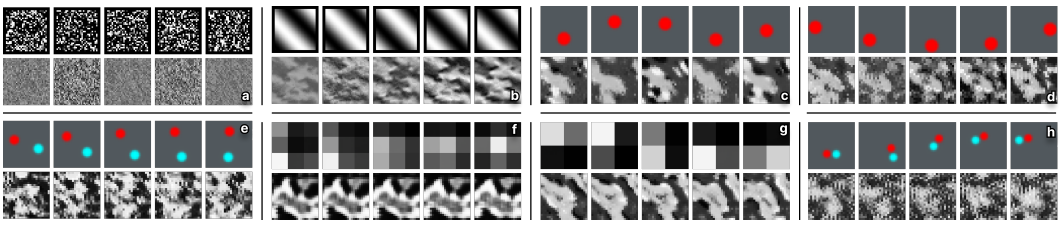


Figure 6.2: Plot of different datasets used in this work (top) and the associated learned hidden state dynamics (bottom). We see the NWM learns different spatiotemporal structure for each dataset, and no structure when trained on random noise (a). Additional videos of dynamics, and code for experiments, can be found at: github.com/akandykeller/NeuralWaveMachines.

Methods — All datasets used in this paper will be considered as unsupervised unless otherwise noted, and thus we will train the model from Section 6.3 as an autoregressive model. To do this, we add a learned decoder from the hidden state $\mathbf{x}_t$ back to the input at the next timestep $\mathbf{u}_{t+1}$, and train the model with a mean-squared error loss. Explicitly, $\hat{\mathbf{u}}_{t+2} = g_\theta(\mathbf{x}_{t+1})$, and $\mathcal{L} = ||\hat{\mathbf{u}}_{t+2} - \mathbf{u}_{t+2}||_2^2$, where g_θ is the decoder which can again be a convolutional neural network, or any network which maps from the spatial hidden state back to the input space. For the simple tasks in Section 6.4, and the sequence classification tasks of Section 6.4 we use minimal encoders and decoders corresponding to single linear layers or small MLPs. For the more complex physical forecasting tasks of Section 6.4 we use the baseline deep convolutional encoders and decoders defined in the benchmark. As a second minor addition which we observe improves performance on long-term trajectory modeling tasks, we introduce an additional encoder network which learns to predict the initial conditions $\mathbf{x}_0$ and $\mathbf{v}_0$ of the network given a partial 'inference' sequence. Explicitly, we can write this as: $\mathbf{x}_0, \mathbf{v}_0 = f_\theta^{IC}(\{\mathbf{u_t}\}_{t=0}^{T_{inf}})$. Such an initial-condition network is common in the Neural-ODE literature (R. T. Q. Chen et al., 2018), and in this setting it is beneficial to initialize the latent dynamics which would otherwise take a significant number of iterations to reach their final magnitude.

Datasets — To investigate how the NWM's representations change when modeling different datasets, we focus on three training sets in this study. Most simply, we first use a dataset of oriented sine functions (depicted in Figure 6.2 b) with a slowly progressing phase over time steps. This dataset is meant to be a very rough approximation to the spontaneously generated retinal waves observed during development (Ackman et al., 2012). For this dataset, the wavelength and magnitude of the sine waves are fixed, and sequences are generated by randomly sampling an orientation between 0 to π and then sequentially progressing the phase by $\frac{1}{9}\pi$ for each

timestep until two periods are complete. As a second dataset, we borrow the rotating MNIST dataset as used in Chapter 5, consisting of sequences of MNIST digits with each timestep rotated by an additional $\frac{1}{9}\pi$ radians. This dataset serves to allow us to investigate the existence of generalizable spatio-temporal structure in a limited setting. Finally, for more realistic dynamics, we make use of the recent hamiltonian dynamics suite (Botev et al., 2021). At a high level, the benchmark consists of a diversity of tasks governed by known equations of motion, including toy physics examples such as idealized springs, pendulums, orbits, and double-pendulums (Fig 6.2 c, d, e & h), as well as cyclic games (f & g). Models are evaluated based on their ability to accurately forecast dynamics into the future from a limited number of inference frames.

Measuring Spatiotemporal Structure

To measure the spatiotemporal representational structure that the NWM learns, and its alignment with natural structure, we start with the two simplest tasks: modeling simple sine waves, and modeling rotating MNIST digits. We use three separate methods for analyzing the representations learned on these tasks: Cohen's d selectivity metric (J. Cohen, 1988) to depict spatial organization, the Hilbert transform to measure the instantaneous phase and velocity of putative waves (Davis, Muller, et al., 2020), and *artificially induced* traveling waves combined with visualized reconstructions to measure the approximate equivalence of latent traveling waves with observed transformations.

Topographic Orientation Selectivity — One of the most common methods to demonstrate spatial organization of neural representations is by measuring their selectivity with respect to different features and plotting this with respect to each neuron's position (David H. Hubel and Torsten N. Wiesel, 1974b). As an initial test of a basic form of selectivity, namely orientation selectivity, we consider a hypothesis from the literature about how such structure might arise initially in animals (Ackman et al., 2012). Specifically, we investigate whether simple periodic inputs, such as the spontaneous retinal waves observed during early development, are sufficient to encourage smooth topographic organization of orientation selectivity when modeled by a minimal NWM. To test this, we train our model on the simple sine waves dataset, and measure the orientation selectivity of each hidden neuron's time-averaged response to a static 36-element sequences of oriented gratings using Cohen's d metric (J. Cohen, 1988). In Figure 6.3 we plot the resulting color/angle of maximal d

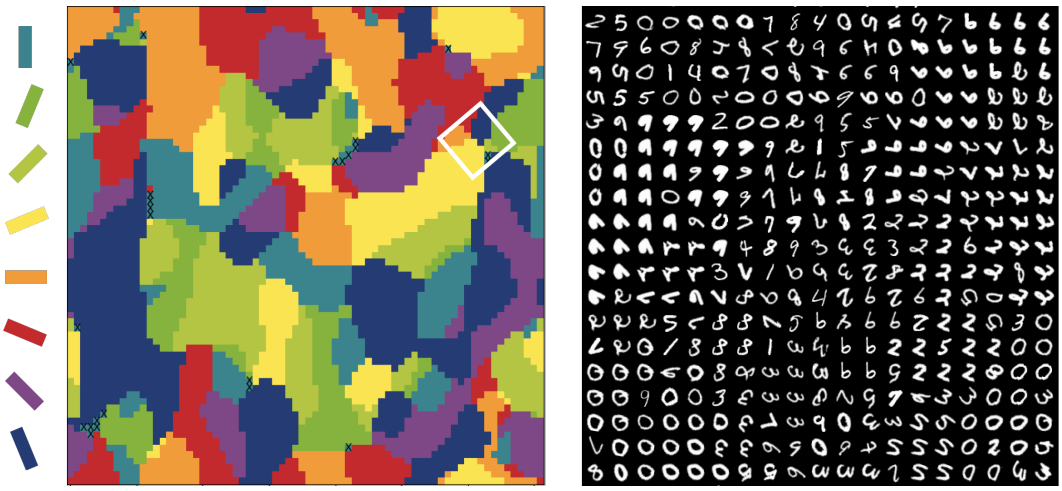


Figure 6.3: (Left) Plot of orientation selectivity of each NWM hidden neuron $\mathbf{x}$ after training on simple sine waves. (Right) Plot of the maximum activating image for a subset of NWM hidden neurons after training on the rotating MNIST dataset (See Sec. C.3 for full). We see the NWM learns smooth spatial topographic structure tailored to the input dataset.

value for each of the 72×72 neurons (or a black $\mathbf{x}$ if all $d < 0.65$). We see that the simulated retinal waves do appear to induce topographic organization of orientation selectivity with superficial similarity to the orientation columns of primary visual cortex (David H Hubel, Torsten N Wiesel, and Stryker, 1978). Outlined in white, we show a manually identified 'pinwheel' where selectivity for all orientations meet, a hallmark of early visual system organization in many species. In relation to prior models of orientation columns (Swindale, 1982), our work does not presuppose the existence of orientation selectivity, but rather it is absent at initialization and it is instead learned in conjunction with topographic organization. We note that the exact statistics of our learned orientation maps have not been measured, and therefore may differ in their current form from those measured in animal studies (Kaschube et al., 2010). In Appendix C.3 we include additional results studying formation mechanism of this orientation selectivity as well as the model parameters which affect the typical length scale of the columns. We leave further precise investigation of the biological similarity to future work.

General Topographic Organization — On the right of Figure 6.3, we show the spatial structure of feature selectivity for a network trained on rotating MNIST digits instead. Specifically, we plot the image from the MNIST dataset which maximally activates each neuron in our 2-dimensional hidden state (at the final timestep). We see that neurons are organized with respect to digit class and style, but also orientation, implying that activity is likely to travel over these paths as a traveling

wave for observed rotation transformations. Such structure is reminiscent of the higher level category selectivity of the higher visual cortices studied in Chapter 4 (Kanwisher, McDermott, et al., 1997; Khosla et al., 2022), and also the temporal structure observed to be related to theta oscillations and waves in the hippocampus (Lubenov and Siapas, 2009).

Instantaneous Phase and Velocity — Next, we demonstrate that the proposed model indeed exhibits full spatiotemporal structure beyond static spatial structure. Compared with biological neural networks, it is easy for us to directly visualize the spatio-temporal activity of our network and qualitatively validate the existence of structure. Figures 6.1, 6.2, and 6.4 provide such examples, while additional samples can be found in Appendix C.3 and the github repository. For additional rigor, however, we borrow state of the art methods from neuroscience to directly compute the instantaneous phase and velocity of putative waves from noisy real-valued signals. Specifically, we follow the work of (Davis, Muller, et al., 2020) and compute the 'generalized phase' of a real valued signal $\mathbf{x}(t)$ by first transforming the signal to a complex-valued analytic signal $\mathbf{x}_a(t)$ through the Hilbert transform $\mathcal{H}$ and then taking the complex argument of this signal as the phase $\phi(t)$ at each point in space and time. Formally: $\mathbf{x}_a(t) = \mathbf{x}(t) + i\mathcal{H}[\mathbf{x}(t)]$, and $\phi(t) = Arg[\mathbf{x}_a(t)]$. Finally, wave velocities can then straightforwardly be computed using the spatial gradient of this phase: $\nu = -\nabla\phi$. In Figure 6.4 we depict such phases and velocities for the NWM trained on the rotating MNIST task. We see that, in alignment with expectation, the estimated phases have a spatially periodic pattern which oscillates with sequence length, while the estimated velocities similarly align to point in the downward direction after training (but not before training, as outlined by the disjoint velocity vectors in Figure 6.4 top right).

Controlled Generation with Induced Traveling Waves — One of the benefits of structured representations in generative models is that they allow for controlled generation of new observations by taking advantage of the known latent operator for a desired input transformation. In this section we demonstrate that such controlled generation is indeed similarly possible by artificially inducing traveling waves in the NWM hidden state, thereby evidencing the spatiotemporal structure of its representations. Given the high degree of flexibility of the potentially emergent wave dynamics of the 2-D system presented in Figure 6.4, we concede that two restrictions must be placed on the model in order for us to be able to accurately induce waves which match those the model has learned. Explicitly, we first define the latent space to be a set

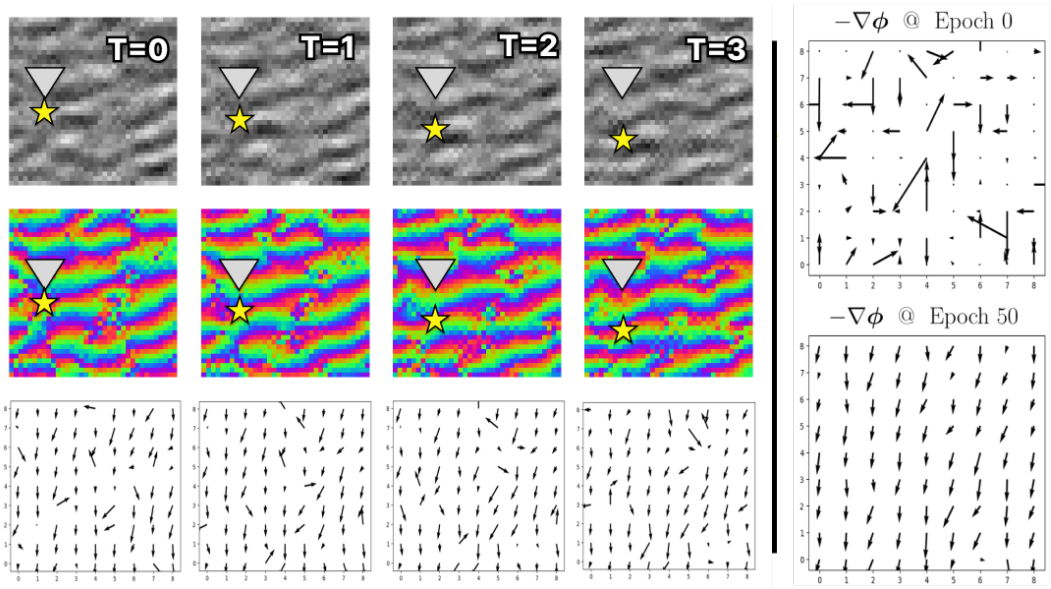


Figure 6.4: (Left) Plot of hidden state $\mathbf{x}$ (top), generalized phase ϕ (mid), and estimated wave velocity $-\nabla\phi$ (bot) over the course of a transformation sequence $T = 0$ to 3. A small gold star moves along with a wave front, relative to a stationary grey triangle, both added to help track the approximate peak of a traveling wave in the hidden state. (Right) Estimated wave velocity before and after training.

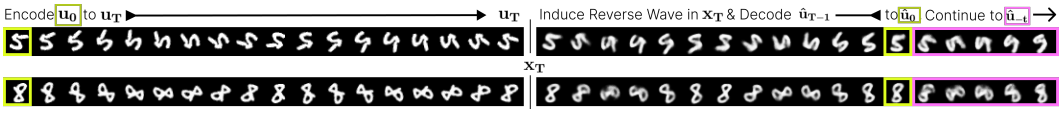


Figure 6.5: Visualization of controlled generation with induced traveling waves. An input sequence from $\mathbf{u}_0$ to $\mathbf{u}_T$ (left) gets encoded to a hidden state $\mathbf{x}_T$. We then induce a traveling wave in the opposite direction of the estimated instantaneous velocity and observe we can decode back to the original input $\hat{\mathbf{u}}_0$ (highlighted yellow, right). Furthermore, we see by continuing the wave, we can continue the transformation past the bounds of the input sequence (highlighted pink, right).

of disjoint 1-dimensional tori such that learned wave propagation will be restricted to a single axis. Secondly, we restrict our topographic coupling to be 1-directional by masking out all weights except for one (non-central weight) in our convolutional kernel which is shared over all tori. In combination, these restrictions ensure that *if* traveling waves are learned by the model, they will likely be able to be approximately modeled by solutions to the 1-dimensional 1-way wave equation: $y(x,t) = f(x-vt)$.

In Figure 6.5 we depict the results of this experiment. In detail, we train the 1D NWM described above on a dataset of length $T = 18$ sequences of rotating MNIST digits. At test time, we encode a full sequence (left) and take the final hidden state $\mathbf{x}_T$ as the initial state for our system. We then *induce a traveling wave* in the hidden state in

the reverse direction of the instantaneous velocity. In practice, since we have limited our system to 1-dimensional tori, this corresponds to sequentially cyclically shifting (or linearly interpolating) activations across the spatial dimension of each circular subspace according to the inverse of our assumed velocity. The result in Figure 6.5 (right) shows that indeed by inducing such reverse traveling waves we can then decode the original input sequence, and even predict elements before the start of the sequence (highlighted in pink). Such sensible decodings highlight the generalization power of the representational structure learned by the NWM. In this example we propagate waves with assumed velocity $v = 1$ and observe that this is slightly faster than the ground truth transformation, resulting in a return to the start state in 14 steps rather than 18. Additional transformations can be found in Appendix Figure C.6.

Computational Implications of Structure

Given the structure measured in Section 6.4 is known to be related to beneficial inductive biases (Fukushima, 1980; T. Cohen and Max Welling, 2016b), in this section we perform preliminary experiments to measure such potential benefits in the context of sequence modeling.

An Inductive Bias for Simple Physical Dynamics — First, inspired by the literature relating traveling waves to visual motion perception (Jancke et al., 2004) and spatiotemporal structure in the hippocampus (Lubenov and Siapas, 2009), we hypothesize that the spatiotemporal structure of the NWM demonstrated in Section 6.4 may serve as an inductive bias towards simple physical dynamics. To measure this, we train NWM models on a representative subset of the Hamiltonian dynamics suite, and measure their error when attempting to forecast long test trajectories into the future. Specifically, we consider six distinct dynamic modeling tasks: three simple physical dynamics including the pendulum, spring, and two body gravitational tasks; one less physical but still temporally smooth task, namely the matching pennies task; and the last, the double pendulum, a complex chaotic physical dynamics task. We compare performance of the NWM with the state of the art baselines using optimal hyperparameters directly given in prior work (Botev et al., 2021; Higgins, Wirnsberger, et al., 2021). These include the HGN++ (Higgins, Wirnsberger, et al., 2021), a standard autoregressive model (AR) (Hochreiter and Schmidhuber, 1997), and a Neural ODE (R. T. Q. Chen et al., 2018) trained both forwards and backwards in time (ODE [TR]). We additionally include a final globally coupled coRNN baseline with equivalent parameters to our NWM to study the direct impact

of the imposed structure on model performance. In Table 6.1 we see that, in alignment with our intuition, the NWM models achieve the lowest forecasting error on the simple physical dynamics tasks, providing evidence in support of the hypothesis that the observed spatiotemporal structure of Section 6.4 is beneficial for modeling such systems. Further, we see that the coRNN baseline performs the best on the less physical but predictable matching pennies task, while the maximally flexible Neural ODE performs the best on the chaotic double pendulum task. Despite these promising results, we note that accurately measuring forecasting performance in image space is notoriously hard (Botev et al., 2021; Higgins, Wirnsberger, et al., 2021), and therefore recommend future work pursue the development of alternative benchmarks and metrics for evaluating the beneficial inductive biases present in the NWM and other forecasting models. In Appendix C.3 and the limitations section below we include additional discussion of these considerations.

Table 6.1: Forward extrapolation mean squared reconstruction error on the Hamiltonian Dynamics Benchmark held-out test set (displayed in units of 1×10^{-8}). We see, in alignment with intuition, the 1 and 2-dimensional Neural Wave Machines (NWM 1D & 2D) perform best on simple physically realistic dynamics such as the spring, pendulum, and two body problem. The globally coupled coRNN performs best on the smooth, but non-physical, matching pennies task, while the maximally flexible Neural ODE performs best on the highly complex and chaotic double pendulum task.

	AR	HGN++	ODE	coRNN	NWM 2D	NWM 1D
Spring	20.97	1.58	1.58	2.52	5.46	**1.45**
Pendulum	4,208.0	166.5	166.0	548.0	**110.9**	237.2
Two Body	91.4	5.0	4.2	2.0	1.9	**0.9**
Pennies	126.3	190.0	119.3	**28.2**	47.2	43.1
Double Pendulum	3,905.0	1,531.0	**1,296.0**	1,666.0	2,512.0	2,821.0

Efficiency — As a second potential benefit related to the NWM's demonstrated spatiotemporal structure, our neural wave machines are highly parameter efficient by design when compared to the globally coupled coRNN. As explained in 6.3, the recurrent connections of our model are restricted to be entirely local as implemented by the convolution operation, thereby allowing for arbitrarily large hidden state sizes with a constant number of recurrent parameters, significantly improving over the quadratically increasing number of parameters in the coRNN. In Table 6.2 we see that on the canonical long sequence classification tasks of sequential MNIST (sMNIST) and permuted sequential MNIST (psMNIST) (T. Konstantin Rusch and Mishra, 2021a), our model achieves comparable performance with the coRNN (and

thus existing state of the art) while requiring a fraction of the parameters. In Appendix C.3 we include additional results on other sequence modeling tasks such as IMDB sentiment classification and long sequence addition showing the same benefits. Interestingly, efficiency in terms of wiring length is also implicated in the formation of orientation columns in natural systems (Koulakov and Chklovskii, 2001). We believe that our work reinforces this relationship from another perspective by showing that when a recurrent oscillatory computational system is constrained to be wiring length efficient by design, it naturally learns topographic organization (e.g. Figure 6.3) in order to optimally function.

Table 6.2: Test accuracy on supervised sequence benchmarks. All results are mean ± std. over 3 random initalizations.

	sMNIST		psMNIST	
	Acc.	$\#\theta$	Acc.	$\#\theta$
coRNN	99.1 ± 0.1	134k	95.0 ± 2.4	134k
NWM	98.6 ± 0.3	50k	94.8 ± 1.1	50k

6.5. Discussion

In this chapter we introduce the Neural Wave Machine, a recurrent neural network model shown to learn spatiotemporally structured representations through local connectivity and oscillatory dynamics. We propose this model as a rich testing ground for the diversity of computational hypotheses surrounding traveling waves in the neuroscience literature, and demonstrate its potential value in this regard by providing evidence for a variety of hypotheses, including one relating to the origin of orientation columns, and one relating to a simple physical inductive bias. Further, we show that this model is competitive with state of the art on sequence modeling tasks, hoping to encourage future use of such models to study the computational purpose of spatiotemporal dynamics in natural systems.

Related Work — In recent years, multiple works have studied the temporal aspects of neural activations and attempted to integrate such structure into deep neural networks. For example, researchers have studied the integration of recurrence into feed forward classification networks (Kietzmann et al., 2019), or the integration spike-time coding through complex activations (Löwe, Lippe, et al., 2022). Separately, others have aimed to directly integrate natural architectural biases by fixing early layers of a convolutional neural network to mimic the early stages of the natural vi-

sual stream, ultimately resulting in improved robustness (Dapello et al., 2020). Our work is highly related to these efforts in motivation, but largely unique in terms of methodology and its focus on complex spatiotemporal dynamics such as traveling waves. One class of models which shares some relation intuitively is reservoir computing (Lukoševičius and Jaeger, 2009). A primary difference between the NWM and reservoir computing frameworks is that our network has a significant number of learned parameters within its recurrence that mediate complex hidden dynamics, while prior work typically relies on a reservoir of fixed dynamics.

Limitations — In this chapter we have put significant effort into quantifying the existence of complex spatiotemporal structure and its impact on the NWMs computational performance. However, due to the inherent flexibility of the possible dynamics which may emerge, there remain limitations in our ability to do so. In future work, we would hope to be able to get a more concrete metric corresponding to spatiotemporal structure to better correlate the structure of our models with their performance. Furthermore, on tasks such as forecasting dynamics, it is still an open question how to best compare the performance of such models in the most comprehensive and fair manner (Higgins, Wirnsberger, et al., 2021). In Appendix C.3 we include additional metrics evaluating model performances on the Hamiltonian Dynamics Suite, highlighting this challenge. Finally, our explorations of parameter efficiency are inherently preliminary and use fully connected encoders and decoders in the NWM, ultimately contributing 45k of the 50k parameters noted for the NWM in Table 6.2. If we were able to replace these components with similarly locally connected functions, such as convolutional networks, the parameter efficiency would further dramatically increase.

Conclusion — As a flexible computational model of traveling waves, we believe the NWM framework offers significant potential to the computational neuroscience community as a method for testing other computational hypotheses relating to traveling waves and synchronous neural dynamics broadly. Similar to convolutional neural networks for modeling the visual system (Daniel L. K. Yamins et al., 2014; Cadieu et al., 2014; Kanwisher, Khosla, et al., 2023), neural wave machines do not match all biologically relevant details of neural dynamics, but we believe they may capture sufficient abstract properties to be useful for performing investigations that otherwise wouldn't be possible. Examples of initial hypotheses which we believe would be primarily suited for future study would be the use of traveling waves as a short term memory mechanism (Bhattacharya, Brincat, et al., 2022), or as a

mechanism for sequencing actions (N. Sato, 2022). In the following chapter we will precisely study the first of these hypotheses, showing that a minimal version of the NWM is capable of storing memories over significantly longer timespans than wave-free counterparts. Ultimately, we believe this work suggests that complex spatiotemporal dynamics and structure should be investigated further in the future to develop the next set of inductive biases necessary to bring deep neural networks to the same levels of efficiency and robustness that we see in natural intelligence.

Chapter 7

TRAVELING WAVES AS AN ENCODING OF THE RECENT PAST

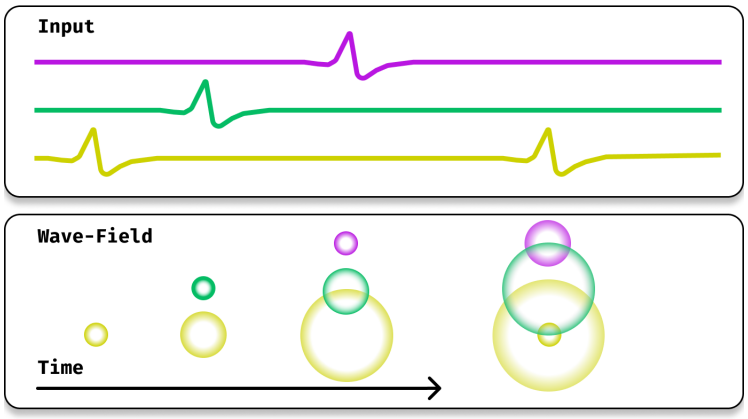


Figure 7.1: Illustration of three input signals (top) and a corresponding wave-field with induced traveling waves (bottom). From an instantaneous snapshot of the wave-field at each timestep we are able decode both the time of onset and input channel of each input spike. Furthermore, subsequent spikes in the same channel do not overwrite one-another.

7.1. Introduction

In the previous chapter we introduced a deep neural network model which exhibits traveling waves of activity in its latent state. We showed that this form of spatiotemporal representational structure served to organize the latent space of such networks, and further served to act as a flexible learned latent operator which was then beneficial for generalization and controlled generation.

In this chapter, we ask what other benefits such representational structure might afford. Specifically, we consider other hypotheses for traveling waves in the literature, and seek to determine if there may be additional benefits to such robust spatiotemporal dynamics. In the context of recurrent neural networks, one hypothesis which is particularly appealing, derived from the study of physical wave fields (Perrard et al., 2016), suggests that traveling waves may serve to efficiently encode recent sequential inputs (Muller, Chavane, et al., 2018). To date however, as with the traveling wave theories of the previous chapter, it has been challenging to test these hy-

potheses due to a lack of standard artificial neural network architectures shown to exhibit such wave-like dynamics.

In the following, we introduce an increasingly minimal recurrent neural network model that exhibits traveling waves of activity within its latent space and test several computational hypotheses. Specifically, we use a suite of synthetic memory tasks to show that models exhibiting traveling waves can solve these tasks orders of magnitude more quickly, do so more accurately, and are able to handle significantly longer sequences than their matched wave-free counterparts. To measure the extent to which this wave-based memory results in generalized performance benefits beyond pathological synthetic tasks, we test our model on modern long-sequence modeling tasks and show that it indeed outperforms wave-free counterparts and is comparable or better than more complex gated recurrent networks such as LSTMs and GRUs. At its core, our work offers two main contributions: first, a simple recurrent neural network architecture that exhibits traveling waves admissible to computational and neuroscientific investigation in a task-oriented manner; and second, a demonstration that traveling waves efficiently encode the recent past thereby benefiting performance on long-sequence tasks.

7.2. Traveling Waves in Recurrent Neural Networks

In this section, we outline how to integrate traveling wave dynamics into a simple recurrent neural network architecture and provide preliminary analysis of the emergent waves.

Simple Recurrent Neural Networks. — As mentioned, the primary goal of this work is to analyze the computational implications of traveling wave dynamics on artificial neural network architectures. In order to reduce potential confounders in this analysis, we strive to study the simplest possible architecture which exhibits traveling waves in its hidden state. To accomplish this, we start with a simple recurrent neural network also known as an Elman Network (Elman, 1990) and consider how we may define its recurrence in order to bias its hidden dynamics towards a simple wave equation. For an input sequence $\{\mathbf{x}_t\}_{t=0}^{T}$ with $\mathbf{x}_t \in \mathbb{R}^d$, and hidden state $\mathbf{h}_0 = \mathbf{0}$ & $\mathbf{h}_t \in \mathbb{R}^n$, a simple RNN (sRNN) is defined with the following recurrence:

$$\mathbf{h}_{t+1} = \sigma(\mathbf{U}\mathbf{h}_t + \mathbf{V}\mathbf{x}_t + \mathbf{b}) \tag{7.1}$$

In such a model, the input encoder and recurrent connections are both linear, i.e. $\mathbf{V} \in \mathbb{R}^{n\times d}$ and $\mathbf{U} \in \mathbb{R}^{n\times n}$, where n is the hidden state dimensionality and σ is a

nonlinearity. The output of the network is then given by another linear map of the final hidden state: $\mathbf{y} = \mathbf{W}\mathbf{h}_T$, with $\mathbf{W} \in \mathbb{R}^{o \times n}$.

Discrete Traveling Waves. — To understand our goal for the time-dynamics of $\mathbf{h}$, we start with the simplest equation which defines our desired dynamics, the one-dimensional one-way wave equation:

$$\frac{\partial h(x,t)}{\partial t} = \nu \frac{\partial h(x,t)}{\partial x} \tag{7.2}$$

Where t is our time coordinate, and x defines the continuous spatial coordinate of our hidden state. Since we wish these waves to propagate through an artificial neural network with discrete neurons and discrete timesteps, we first must discretize Equation 7.2 in both space and time. If we define our hidden neurons to be laid out on a one-dimensional line at regular intervals, we see that the neuron at location x, i.e. $h(x,t)$ is equivalent to the x'th neuron: h_t^x. In practice, we define our neurons to be arranged in a one-dimensional circle (S^1) to avoid boundary effects. Then, if we assume a velocity $\nu = 1$, we see that we can write a discretization of this equation for small Δt as: $h_{t+\Delta t}^x = h_t^{x+\Delta x}$. In matrix form, this is equivalent to multiplication of the hidden state vector with a circular shift matrix:

$$\mathbf{h}_{t+1} = \Sigma \mathbf{h}_t \quad \text{where} \quad \Sigma = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{7.3}$$

We thus see that starting from the sRNN in Equation 7.1, there are three components we need to consider to reach solutions approaching Equation 7.3: the activation function σ, the recurrent connectivity $\mathbf{U}$ and proper initialization. In the following we will outline the choices for each which ultimately define what we call the Wave-RNN (wRNN).

Activation Functions. — Common choices for recurrent neural network activation functions include logistic functions (sigmoid), hyperbolic tangent functions (tanh), and rectified linear functions (ReLU). Prior work studying simple recurrent neural networks has found that linear and rectified linear activations have theoretical and empirical benefits for long-sequence modeling (Q. V. Le et al., 2015; Orvieto et al., 2023). In this work, we define σ = ReLU in order to align with Equation 7.3 and

this prior work. Empirically we also find this to be significantly more performant than $\sigma = \tanh$.

Recurrent Connectivity. — In pursuit of the goal of equating equations 7.1 and 7.3, we see that defining the matrix $\mathbf{U}$ to be a randomly initialized dense matrix is likely to be detrimental to the emergence of waves given the desired diagonal structure of Σ. However, a common linear operator which has a very similar diagonal structure to Σ is the convolution operation. Specifically, assuming a single input channel, and a length 3 convolutional kernel $\mathbf{u} = [0, 0, 1]$, we see the following equivalence:

$$\mathbf{u} \star \mathbf{h}_{t-1} = \Sigma \mathbf{h}_{t-1} \tag{7.4}$$

where $\star$ defines circular convolution over the hidden state dimensions n. In practice we find that increasing the number of channels helps the model to learn significantly faster and reach lower error. To do this, we define $\mathbf{u} \in \mathbb{R}^{c \times c \times f}$ where c is the number of channels, and f is the kernel size, and we reshape the hidden state from a single n dimensional circle to c separate $n' = \lfloor \frac{n}{c} \rfloor$ dimensional circular channels (e.g. $\mathbf{h} \in \mathbb{R}^{c \times n'}$). We can then write the full recurrence as:

$$\mathbf{h}_{t+1} = \sigma(\mathbf{u} \star \mathbf{h}_t + \mathbf{V}\mathbf{x}_t + \mathbf{b}) \tag{7.5}$$

where $\mathbf{V}\mathbf{x}_t$ is similarly reshaped to match the channel structure of $\mathbf{h}$.

Initialization. — Finally, similar to the prior work with recurrent neural networks (Q. V. Le et al., 2015; Gu, Goel, et al., 2022), we find careful choice of initialization can be crucial for the model to converge more quickly and reach lower final error. Specifically, we initialize the convolution kernel such that the matrix form of the convolution (known as a Toeplitz matrix) is exactly that of the shift matrix Σ for each channel separately. Succinctly, in the PyTorch framework (Paszke et al., 2019), this initialization can be implemented as:

```
# Shift initalization for U
U = torch.zeros(size=(c, c, kernel_size))
torch.nn.init.dirac_(U)
U = torch.roll(input=U, shifts=1, dims=-1)
```

Furthermore, we find that initializing the matrix $\mathbf{V}$ to be zero with a single identity mapping from the input to a single hidden unit to further drastically improve training speed. Again, explicitly:

```
# Sparse identity initialization for V
```

```
V = torch.zeros(size=(n_hidden, n_input))
v = V.view(c, n_hidden // c, n_inp)
v[:, 0] = 1.0
```

Intuitively, these initalizations combined can be seen to support a separate traveling wave of activity in each channel, driven by the input at a single source location.

Wave Recurrent Neural Networks. — Combining the above ReLU activation function, convolutional recurrent connections, and specific initalizations, we reach our definition of the Wave-RNN. We note that these are not the only choices which lead to wave-dynamics in the hidden state, however empirically we find them to be the most performant and also the most conducive to wave activity. At the end of Section 7.3, we empirically demonstrate this through a range of ablation experiments.

Baselines. — In order to isolate the effect of traveling waves on model performance, we desire to pick baseline models which are as similar to the Wave-RNN as possible while not exhibiting traveling waves in their hidden state. To accomplish this, we rely on the Identity Recurrent Neural Network (iRNN) of Q. V. Le et al. (2015). This model is nearly identical to the Wave-RNN, constructed as a simple RNN with σ = ReLU, but uses an identity initialization for $\mathbf{U}$. Despite its simplicity the iRNN is found to be comparable to LSTM networks on standard benchmarks, and thus represents the ideal highly capable simple recurrent neural network which is comparable to the Wave-RNN.

Analysis of Traveling Waves. — Before we study the memory and sequence modeling capabilities of the proposed architecture, we first demonstrate that the model does indeed produce traveling waves within its hidden state. To do this, in Figure 7.2, for the best performing (wRNN & iRNN) models on the Sequential MNIST task of the proceeding section, we plot in the top row the activations of our neurons (vertical axis) over time (horizontal axis) as the RNNs process a sequence of inputs (MNIST pixels). As can be seen, there are distinct diagonal bands of activation for the Wave-RNN (left), corresponding to waves of activity propagating between hidden neurons over time. For the baseline simple RNN (iRNN) right, no such bands exist, but instead stationary bumps of activity exist for durations of time and then fade. In the bottom row, following the analysis techniques of Davis et al. (2021) (Davis, G. B. Benigno, et al., 2021), we plot the corresponding 2D Fourier transform of the above activation time series. In this plot, the vertical axis corresponds to spatial frequencies while the horizontal axis corresponds to the usual temporal frequencies.

In such a 2D frequency space, a constant speed traveling wave (or general moving object (Cagigal et al., 1995)) will appear as a linear correlation between spatial and time frequencies. Specifically, the slope of that correlation will then correspond to the speed. Indeed, for the wave RNN, we see a strong band of energy along the diagonal corresponding to our traveling waves with velocity ≈ 0.3 $\frac{\text{units}}{\text{timestep}}$; as expected, for the iRNN we see no such diagonal band in frequency space.

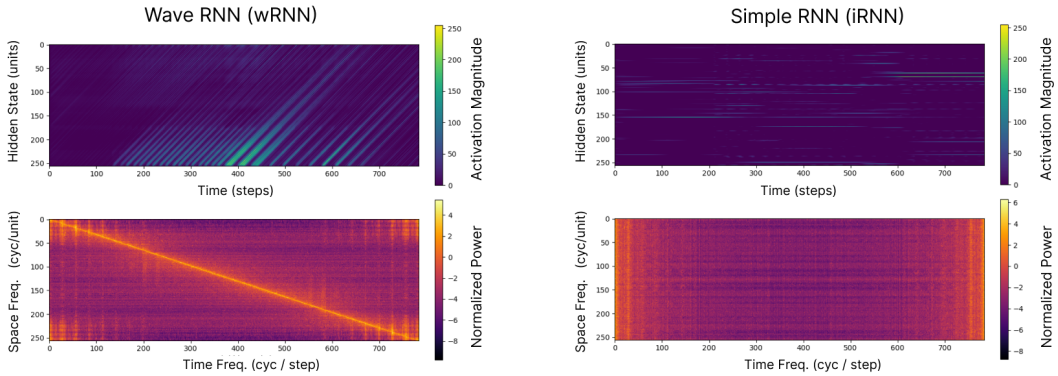


Figure 7.2: Visualization of hidden state (top) and associated 2D Fourier transform (bottom) for a wRNN (left) and iRNN (right) operating on the sMNIST task. We see the Wave-RNN exhibits a clear flow of activity across the hidden state (diagonal bands) while the iRNN does not.

7.3. Experiments

In this section we aim to leverage the model introduced in Section 7.2 to test the computational hypothesis that traveling waves may serve as a mechanism to encode the recent past in a wave-field short-term memory. To do this, we first leverage a suite of frequently used synthetic memory tasks designed to precisely measure the ability of sequence models to store information and learn dependencies over variable length timescales. Following this, we use a suite of standard sequence modeling benchmarks to measure if the demonstrated short-term memory benefits of wRNNs persist in a more complex regime. For each task we perform a grid search over learning rates, learning rate schedules, and gradient clip magnitudes, presenting the best performing models from each category on a held-out validation set in the figures and tables. In the appendix we include the full ranges of each grid search as well as exact hyperparameters for the best performing models in each category.

Copy Task

As a first analysis of the impact of traveling waves on memory encoding, we measure the performance of the wRNN on the standard 'copy task', as frequently employed in prior work (Graves, Wayne, and Danihelka, 2014; Arjovsky, Shah, et al., 2016; Gu, Gulcehre, et al., 2020; Henaff et al., 2016). The task is constructed of sequences of categorical inputs of length $T + 20$ where the first 10 elements are randomly chosen one-hot vectors representing a category in $\{1, \dots 8\}$. The following T tokens are set to category 0, and form the time duration where the network must hold the information in memory. The next token is set to 9, representing a delimiter, signaling the RNN to begin reproducing the stored memory as output, and the final 9 tokens are again set to category 0. The target for this task is another categorical sequence of length $T + 20$ with all elements set to category 0 except for the last 10 elements containing the initial random sequence of the input to be reproduced. At a high level, this task tests the ability for a network to encode categorical information and maintain it in memory for T timesteps before eventually reproducing it. Given the hypothesis that traveling waves may serve to encode information in an effective 'register', we hypothesize that wave-RNNs should perform significantly better on this task than the standard RNN. For each sequence length we compare wRNNs with 100 hidden units and 6 channels ($n = 100, c = 6$) with two baselines: iRNNs of comparable parameter counts ($n = 100 \implies$ 12k params.), and iRNNs with comparable numbers of activations ($n = 625$) but a significantly greater parameter count ($\implies$ 403k params.).

In Figure 7.3, we show the performance of the best performing baseline RNNs and wRNNs, obtained from our grid search, for $T = \{0, 30, 80\}$. We see that the wRNNs achieve more than 5 orders of magnitude lower loss and learn significantly faster for all sequence lengths. From the visualization of the model outputs in Figure 7.4, we see that the iRNN has trouble holding items in memory for longer than 10 timesteps, while the comparable wRNN has no problem copying data for up to 500 timesteps.

Adding Task

To bolster our findings from the copy task, we employ the long-sequence addition task originally introduced by Hochreiter and Schmidhuber (1997). The task consists of a two dimensional input sequence of length T, where the first dimension is a random sample from $\mathcal{U}([0, 1])$, and the second dimension contains only two non-zero elements (set to 1) in the first and second halves of the sequence respectively.

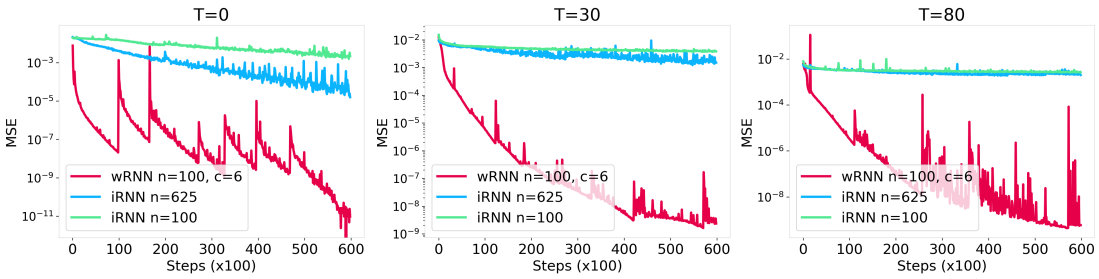


Figure 7.3: Copy task with lengths T={0, 30, 80}. wRNNs achieve > 5 orders of magnitude lower loss.

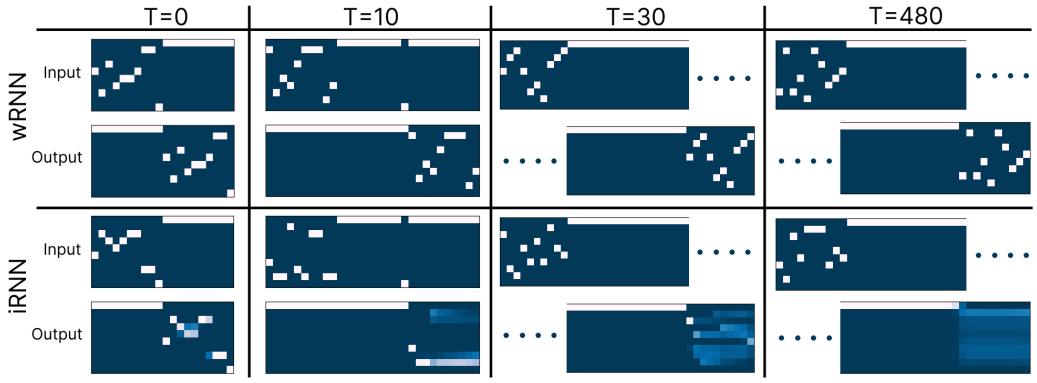


Figure 7.4: Examples from the copy task for wRNN (n=100, c=6) and iRNN (n=100). We see the iRNN loses significant accuracy after T=10 while the wRNN remains perfect at T=480 (MSE $\approx 10^{-9}$).

The target is the sum of the two elements in the first dimension which correspond to the non-zero indicators in the second dimension. Similar to the copy task, this task allows us to vary the sequence length and measure the limits of each model's ability.

The original iRNN paper (Q. V. Le et al., 2015) demonstrated that standard RNNs without identity initialization struggle to solve sequences with $T > 150$, while the iRNN is able to perform equally as well as an LSTM, but begins to struggle with sequences of length greater than 400 (a result which we reconfirm here). In our experiments depicted in Figure 7.5 and Table 7.1, we find that the wRNN not only solves the task much more quickly than the iRNN, but it is also able solve significantly longer sequences than the iRNN (up to 1000 steps). In these experiments we use an iRNN with $n = 100$ hidden units (10.3k parameters) and a wRNN with $n = 100$ hidden units and $c = 27$ channels (10.29k parameters).

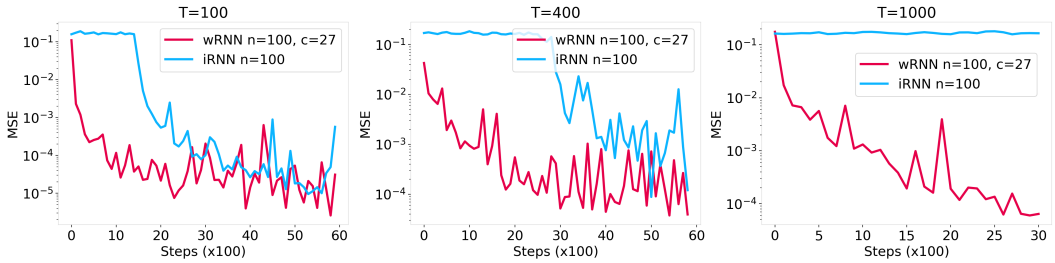


Figure 7.5: wRNN and iRNN Training curves on the addition task for three different sequence lengths (100, 400, 1000). We see that the wRNN converges significantly faster than the iRNN on all lengths, achieves lower error, and can solve tasks which are significantly longer.

	Seq. Length (T)	100	200	400	700	1000
iRNN	Test MSE	1×10^{-5}	4×10^{-5}	1×10^{-4}	0.16	0.16
	Solved Iter	14k	22k	30k	×	×
wRNN	Test MSE	$\mathbf{4 \times 10^{-6}}$	$\mathbf{2 \times 10^{-5}}$	$\mathbf{4 \times 10^{-5}}$	$\mathbf{8 \times 10^{-5}}$	$\mathbf{6 \times 10^{-5}}$
	Solved Iter.	**300**	**1k**	**1k**	**3k**	**2k**

Table 7.1: Long sequence addition task for different sequence lengths. The wRNN finds the task solution (defined as MSE $\leq 5 \times 10^{-2}$) multiple orders of magnitude quicker and is able to solve much longer tasks than the iRNN. The × indicates the model never solved the task after 60k iterations.

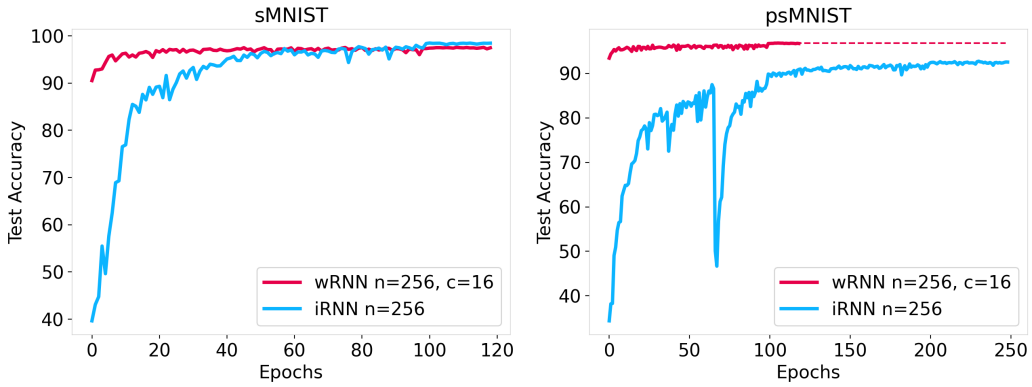


Figure 7.6: sMNIST (left) and psMNIST (right) training curves for the iRNN & wRNN. The wRNN trains much faster and is virtually unaffected by the sequence permutation, while the iRNN suffers.

Sequential Image Classification

Given the dramatic benefits that traveling waves appear to afford in the synthetic memory-specific tasks, in this section we additionally strive to measure if waves will have any similar benefits for more complex sequence tasks relevant to the machine

learning community. One common task for evaluating sequence models is sequential pixel-by-pixel image classification. In this work we specifically experiment with three sequential image tasks: sequential MNIST (sMNIST), permuted sequential MNIST (psMNIST), and noisy sequential CIFAR10 (nsCIFAR10). The MNSIT tasks are constructed by feeding the 784 pixels of each image of the MNIST dataset one at a time to the RNN, and attempting to classify the digit from the hidden state after the final timestep. The permuted variant applies a random fixed permutation to the order of the pixels before training, thereby increasing the task difficulty by preventing the model from leveraging statistical correlations between nearby pixels. The nsCIFAR10 task is constructed by feeding each row of the image (32×3 pixels) flattened as vector input to the network at each timestep. This presents a significantly higher input-dimensionality than the MNIST tasks, and additionally contains more complicated sequence dependencies due to the more complex images. To further increase the difficulty of the task, the sequence length is padded from the original length (32), to a length of 1000 with random noise. Therefore, the task of the model is not only to integrate the information from the original 32 sequence elements, but additionally ignore the remaining noise elements. As in the synthetic tasks, we again perform a grid search over learning rates, learning rate schedules, and gradient clip magnitudes. Because of our significant tuning efforts, we find that our baseline iRNN results are significantly higher than those presented in the original work (98.5% vs. 97% on sMNIST, 91% vs. 81% on psMNIST), and additionally sometimes higher than many 'state of the art' methods published after the original iRNN. In the tables below we indicate results from the original work by a citation next to the model name, and lightly shade the rows of our results.

In Table 7.2, we show our results in comparison with existing work on the sMNIST and psMNIST. Despite the simplicity of our proposed approach, we see that it performs favorably with many carefully crafted RNN and convolutional architectures. We additionally include wRNN + MLP, which is the same as the existing wRNN, but replaces the output map $\mathbf{W}$ with a 2-layer MLP. We see this increases performance significantly, suggesting the linear decoder of the basic wRNN may be a performance bottleneck. In Figure 7.6 (left), we plot the training accuracy of the best performing wRNN compared with the best performing iRNN over training iterations on the sMNIST dataset. We see that while the iRNN reaches a slightly higher final accuracy (+0.9%), the wRNN trains remarkably faster at the beginning of training, taking the iRNN roughly 50 epochs to catch up. On the right of the figure, we plot the models' performance on the permuted variant of the task (psMNIST) and see

Model	sMNIST	psMNIST	n / $\#\theta$
uRNN (Arjovsky, Shah, et al., 2016)	95.1	91.4	512 / 9k
iRNN	98.5	92.5	256 / 68k
LSTM (T Konstantin Rusch et al., 2022)	98.8	92.9	256 / 267k
GRU (T Konstantin Rusch et al., 2022)	99.1	94.1	256 / 201k
IndRNN (6L) (S. Li et al., 2018)	99.0	96.0	128 / 83k
Lip. RNN (Erichson et al., 2021)	99.4	96.3	128 / 34k
coRNN (T. Konstantin Rusch and Mishra, 2021a)	99.3	96.6	128 / 34k
LEM (T Konstantin Rusch et al., 2022)	99.5	96.6	128 / 68k
wRNN 16c	97.6	96.7	256 / 47k
URLSTM (Gu, Gulcehre, et al., 2020)	99.2	97.6	1024 / 4.5M
wRNN + MLP	97.5	97.6	256 / 420k
pLMU (Chilkuri and Eliasmith, 2021)	-	98.5	468 / 165k
FlexTCN (Romero et al., 2022)	99.6	98.63	- / 375k

Table 7.2: sMNIST & psMNIST (sorted) test accuracy.

the performance of the Wave-RNN is virtually unaffected, while the simple RNN baseline suffers dramatically.

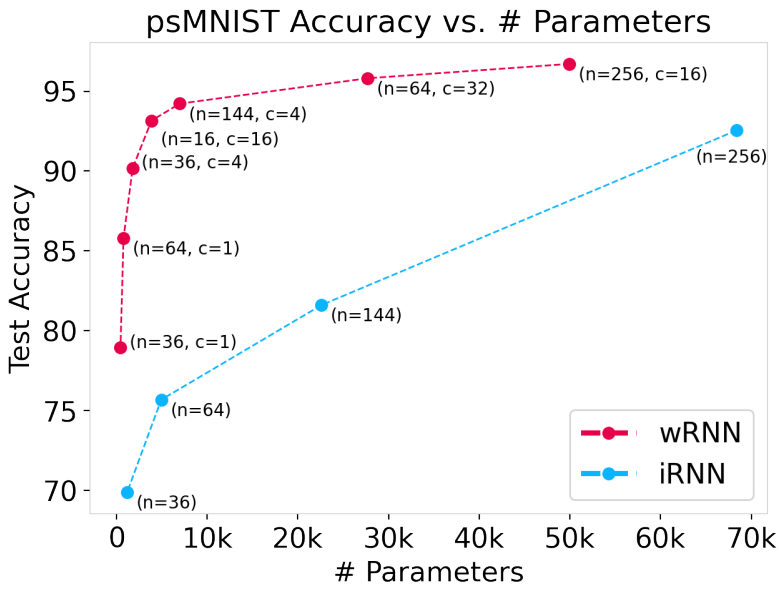


Figure 7.7: Number of parameters vs. accuracy for wRNNs & iRNNs on psMNIST, attained by varying hidden state size (n) and number of channels (c). We see that the wRNN achieves significantly higher accuracy at all levels of parameter counts.

We note that in addition to faster training and higher accuracy, the wRNN model additionally exhibits substantially greater parameter efficiency than the iRNN due

to its convolutional recurrent connections in place of fully connected layers. To exemplify this, in Figure 7.7 we show the accuracy (y-axis) of a suite of wRNN models plotted as a function of the number of parameters (x-axis). We see that compared with the iRNN, the wRNN reaches near maximal performance with significantly fewer parameters, and retains a performance gap over the iRNN with increased parameter counts.

Finally, to see if the benefits of the wRNN extend to more complicated images, we explore the noisy sequential CIFAR10 task. In Figure 7.8 we plot the training curves of the best performing models on this dataset, and see that the Wave-RNN still maintains a significant advantage over the iRNN in this setting. In Table 7.3, we see the performance of the wRNN is ahead of standard gated architectures such as GRUs and LSTMs, but also ahead of more recent complex gated architectures such as the Gated anti-symmetric RNN (Chang et al., 2019). We note that the parameter count for the wRNN on this task is significantly higher than the other models listed in the table. This is primarily due to the linear encoder mapping from the high dimensionality of the input (96) to the large hidden state. In fact, for this model, the encoder $\mathbf{V}$ alone accounts for $> 90\%$ of the parameters of the full model (393k/435k). If one were to replace this encoder with a more parameter efficient encoder, such as a convolutional neural network or a sparse matrix (inspired by the initialization for $\mathbf{V}$), the model would thus have significantly fewer parameters, making it again comparable to state of the art. We leave this addition to future work, but believe it to be one of the most promising approaches to improving the wRNN's general competitiveness. Ultimately, we see that the wRNN performance is significantly improved over the matched wave-free model. Furthermore, we see that it is surprisingly competitive with state of the art long-sequence models despite having no imposed long-term memory inductive biases besides wave-propagation. We believe that these results therefore serve as strong evidence in support of the hypothesis that traveling waves may be a valuable inductive bias for encoding the recent past and thereby facilitate long-sequence learning.

Ablation Experiments

In this section we include a final set of ablation experiments to validate the architecture choices for the Wave-RNN and provide further evidence for the hypothesis that traveling waves are a beneficial inductive bias for sequence learning. For each of the results reported below, a grid search over learning rates, activation functions, inital-

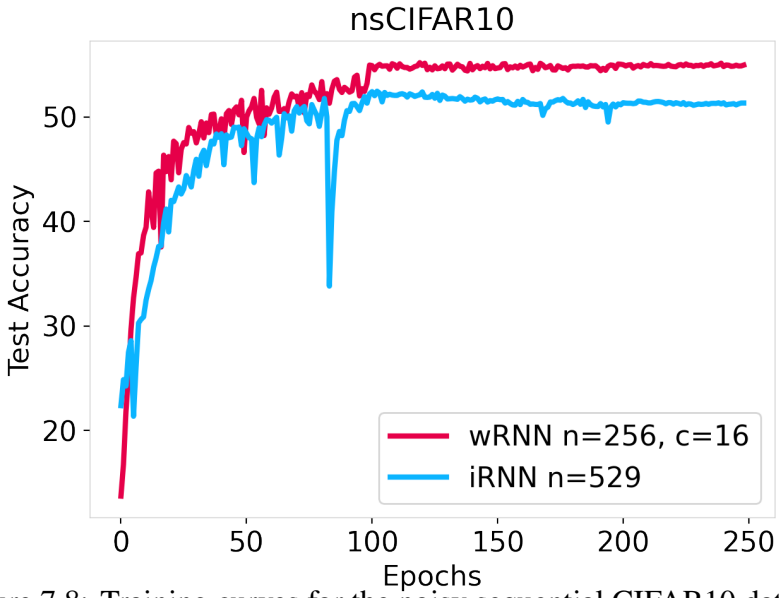


Figure 7.8: Training curves for the noisy sequential CIFAR10 dataset.

Model	Acc.	# units / params
LSTM (T Konstantin Rusch et al., 2022)	11.6	128 / 116k
GRU (T Konstantin Rusch et al., 2022)	43.8	128 / 88k
anti-sym. RNN (Chang et al., 2019)	48.3	256 / 36k
iRNN	51.3	529 / 336k
Incremental RNN (Kag et al., 2020)	54.5	128 / 12k
Gated anti-sym. RNN (Chang et al., 2019)	54.7	256 / 37k
wRNN (16c)	55.0	256 / 435k
Lipschits RNN (Erichson et al., 2021)	57.4	128 / 46k
coRNN (T. Konstantin Rusch and Mishra, 2021a)	59.0	128 / 46k
LEM (T Konstantin Rusch et al., 2022)	60.5	128 / 116k

Table 7.3: Test set accuracy on the noisy sequential CIFAR dataset sorted by performance.

izations, and gradient clipping values was again performed to ensure fair comparison.

In Table 7.4, we show the performance of the wRNN on the copy task as we ablate various proposed components such as convolution, **u**-shift initialization, and **V** initialization (as described in Section 7.2). At a high level, we see that the the **u**-shift initialization has the biggest impact on performance, allowing the model to successfully solve tasks greater than length $T = 10$. We find the **V** initialization to be slightly less impactful, improving final performance only marginally, but mainly significantly increasing the speed of convergence of models (not pictured). In addition to ablating the wRNN, we additionally explore initializing the iRNN with a shift initialization ($\mathbf{U} = \Sigma$) and sparse identity initialization for **V** to disassociate these effects from the effect of the convolution operation. We see that the addition of Σ initialization to the iRNN improves its performance dramatically, but it never reaches the same level of performance of the wRNN – indicating that the sparsity and tied weights of the convolution operation are critical to memory storage and retrieval on this task.

Model	Sequence Length (T)			
	0	10	30	80
wRNN	$\mathbf{9 \times 10^{-12}}$	$\underline{1 \times 10^{-10}}$	$\mathbf{8 \times 10^{-11}}$	$\mathbf{1 \times 8^{-11}}$
- **V**-init	$\underline{1 \times 10^{-11}}$	$\mathbf{2 \times 10^{-11}}$	$\underline{4 \times 10^{-10}}$	$\underline{4 \times 10^{-11}}$
- **u**-shift-init	5×10^{-11}	3×10^{-10}	7×10^{-4}	6×10^{-4}
- **V**-init - **u**-shift-init	8×10^{-10}	1×10^{-4}	3×10^{-4}	7×10^{-4}
iRNN (n=100)	1×10^{-4}	3×10^{-3}	2×10^{-3}	1×10^{-3}
+ Σ-init	1×10^{-8}	1×10^{-7}	2×10^{-7}	2×10^{-5}
+ Σ-init + **V**-init	1×10^{-7}	1×10^{-7}	1×10^{-6}	8×10^{-6}

Table 7.4: Ablation test results (MSE) on the copy task. Best results are bold, second best underlined.

One particularly interesting finding from our ablation studies is that although models without shift initialization do not initially exhibit traveling waves, most randomly initialized models that do learn to perform well on the task eventually learn to exhibit waves in their hidden state. In Figure 7.9 we show the activation sequence for a wRNN model after initializing kernels with the random Kaming Uniform initialization (He et al., 2015), and then training on the sMNIST task. We see that early in training (left) the model does not exhibit traveling waves; however, by the end of training (right), the randomly initialized model has learned to exhibit waves, and ultimately achieves higher test accuracy than a comparable identity initialized

model which never learns to exhibit waves (96.5% vs. 94.8%, training curves in appendix). In the appendix, we include additional ablation studies removing weight sharing in the convolutional layer, and freezing $\mathbf{U}$ & $\mathbf{V}$ at initialization, demonstrating the robustness of traveling waves in these models and providing further support for our empirical conclusions.

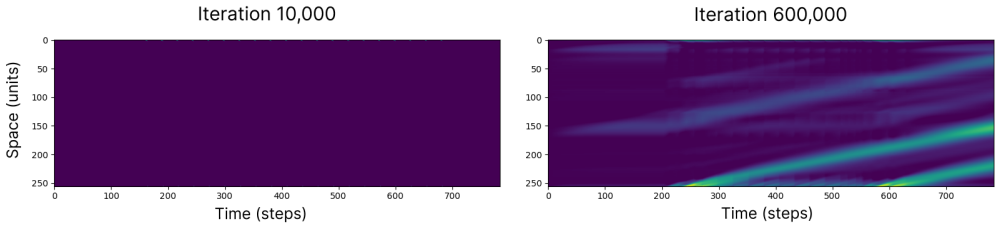


Figure 7.9: Visualization of the hidden state for a wRNN with randomly initialized kernels (Kaming Uniform). We see the model learns to exhibit traveling waves in its hidden state through training.

7.4. Discussion

In this section we include a preliminary discussion of related work with a more complete overview in the appendix. Compared with the Neural Wave Machine introduced in the previous chapter, this present chapter introduces a more minimal model capable of producing traveling waves, and thereby permits the study of the direct implications of traveling wave dynamics compared with non-wave models on standard sequence modeling benchmarks. Because our model is a standard simple RNN, the experimental conclusions resulting from the wave dynamics may generalize to more complex state of the art architectures and similarly enhance their performance.

Y. Chen et al. (2022) demonstrated that in a predictive RNN autoencoder learning sequences, a Toeplitz connectivity emerges spontaneously, replicating multiple canonical neuroscientific measurements such as one-shot learning and place cell phase precession. Our results in Figure 7.9 further support these findings that with proper training and connectivity constraints, recurrent neural networks can learn to exhibit traveling wave activity in service of solving a task.

In another related paper, G. Benigno et al. (2022) studied a complex-valued recurrent neural network with distance-dependant connectivity and signal-propagation time-delay, in order to understand the potential computational role for traveling waves that have been observed in visual cortex. They showed that when recurrent

strengths are set optimally, the network is able to perform long-term closed-loop video forecasting significantly better than networks lacking this spatially-organized recurrence. Our model is complimentary to this approach, focusing instead on the sequence integration capabilities of waves, rather than on forecasting, and leveraging a more traditional deep learning architecture.

Limitations & Future Work.

The experiments in this chapter are inherently limited due to the relatively small scale of the datasets and models studied. For example, nearly all models in this work rely on linear encoders and decoders, and consist of a single RNN layer. Compared with sate of the art models consisting of more complex deep RNNs with skip connections and regularization, our work is therefore potentially leaving significant performance on the table. However, as described above, beginning with small scale experiments on standard architectures yields alternative benefits including more accurate hyperparameter tuning (due to a smaller search space) and potentially greater generality of conclusions. In future work, we plan to test the full implications of our results for the machine learning community and integrate the core concepts from the wRNN into more modern sequence learning algorithms, such as those used for language modeling.

A limitation of our model specifically is the increased number of activations when using a large number of channels. Succinctly, the wRNN requires more memory than the iRNN model in order to store the larger hidden state over time. This limitation can be partially ameliorated by using an adjoint method in the backwards pass so that hidden states can be recomputed when needed and not stored for intermediate steps (R. T. Q. Chen et al., 2018). In preliminary experiments we find that such an approach indeed works and produces waves, offering a suggestion for scaling such models to larger sizes.

Finally, we believe this work opens the door for significant future work in the domains of theoretical and computational neuroscience. For example, direct comparison of the wave properties of our model with neurological recordings may provide novel insights into the mechanisms and role of traveling waves in the brain, analogous to how comparison of visual stream recordings with convolutional neural network activations has yielded insights into the biological visual system (Cadieu et al., 2014).

Chapter 8

LEARNING FACTORIZED REPRESENTATIONS WITH SPATIO-TEMPORAL FLOWS

8.1. Introduction

In the previous chapters, we have introduced multiple methods to induce spatio-temporal structure into artificial neural network representations. At a high level, these methods worked by initially specifying some spatial organization of artificial neurons through a neighborhood structure (such as a 2-D grid or torus), and then encouraging a flow of activity between neighboring neurons over time steps. This flow was either achieved through a specific spatio-temporal prior distribution (in the TVAE), or more flexibly through a set of local recurrent connections (in the NWM). These methods, inspired by group equivariant neural network architectures, were demonstrated to indeed have many beneficial properties for both the machine learning and computational neuroscience communities, and further had a clear correspondence with their group-equivariant counterparts.

However, what if we desired to encourage our neural activity to follow a more precise spatio-temporal structure? For instance, could we leverage the aforementioned approaches to encourage our neural activity to follow an arbitrary partial differential equation (PDE) such as the traditional wave equation: $\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$? As we saw in the previous chapter, we were able to accomplish this successfully for simple ODEs such as the one-way wave equation, however if we were to require higher order spatial derivatives it is likely we may run into issues. Unfortunately, due to the discrete nature of our grids of neurons in the previous chapters, the prior approach has clear limitations when simulated on digital computers. Foremost, the resolution of the grid that we would like to construct is limited by total number of neurons in the final layer. Since adding neurons can be quite computationally expensive, this inherently limits the resulting resolution and sizes of the topologies we can build. As a result, when attempting to approximate a PDE over a low-resolution discretization of space, the approximation errors become significantly large, ultimately dramatically affecting the evolution of the system and pulling it away from the desired dynamics. How then might we be able to induce PDE-like spatio-temporal structure in neural representations without having such computational difficulties?

In the machine learning literature, a new class of models has recently emerged as a popular approach to represent signals over a continuous space efficiently without incurring discretization or scaling challenges. Such models are known as implicit neural representations, and work by learning the parameters θ of a function f_θ which maps from the desired coordinate space directly to the value of the signal (Mildenhall et al., 2021; Z. Chen and H. Zhang, 2019; Sitzmann et al., 2020). For example, if the goal were to represent the height above sea level for each latitude and longitude coordinate on Earth, a function $f_\theta : (\text{lat}, \text{long}) \rightarrow \mathbb{R}$ could be trained to directly output the altitude for each pair of coordinates (lat, long) sampled from the surface of the sphere. In this way, the function f is then effectively infinite resolution since any real valued coordinate pair can be sampled to produce an output. For the settings which interest this thesis, one could imagine similarly parameterizing the surface of the cortex via an implicit neural representation, and learning a function which maps from each coordinate on the cortical surface to a specific signal value. By doing so, we effectively achieve infinite resolution allowing for the precise computation of gradients and divergences of this signal along the cortical surface. The question then becomes what signal should we be representing implicitly?

In this chapter, we propose an approach to induce spatio-temporal structure which diverges quite significantly from that of the previous chapters. Rather than trying to learn a map from the input to neural activations which itself obeys some spatio-temporal structure, we instead propose to learn a set of 'forces' in activation space which move activations in a structured manner in order to accurately model observed transformations. To do this, we infer the initial set of activations $\boldsymbol{z}_0$ with a latent variable model (such as a variational autoencoder), and then simultaneously learn a set of k implicit scalar-valued potential functions defined over that space (which we denote $u^k(\boldsymbol{z}, t)$) which impose forces on the activations through their gradients $\nabla_z u^k$. We are then able to formulate a generative model of sequences which begins with a latent variable $\boldsymbol{z}_0$ and a latent potential index k, and then computes latent variables at future time steps through the gradient: $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \nabla_z u^k(\boldsymbol{z}, t)$.

In order to encourage the function u to obey our desired spatio-temporal structure, we propose to leverage the well known Physics Informed Neural Network (PINN) framework of Raissi et al. (2019). In short, this framework imposes an additional loss on our function $u(\boldsymbol{z}, t)$ which encourages it to satisfy a PDE of our choice. For example, to learn a function u which obeys the wave equation, one could simply add the loss term $||\frac{\partial^2 u}{\partial t^2} - c^2 \nabla_z^2 u(\boldsymbol{z}, t)||_2^2$ to the overall loss during training. Since u is defined

over a continuous space z and parameterized via a neural network, computing the required derivatives is efficient and accurate through automatic differentiation tools.

In the following, we will describe precisely how we define a generative model of sequences using this idea. We will then show how, by learning a set of k distinct potential functions, the model can be seen to be effectively factorizing its latent space according to k separate learned transformations. Due to these transformations being modeled as flows of probability on a potential landscape, we will refer to this model as 'Flow Factorized Representation Learning'. We will show theoretically how these flows can be seen to actually follow the 'optimal transport' path as derived using dynamic optimal transport theory. Finally, we will show empirically how these models then become approximately equivariant with respect to observed transformations, achieving lower equivariance error compared with prior work.

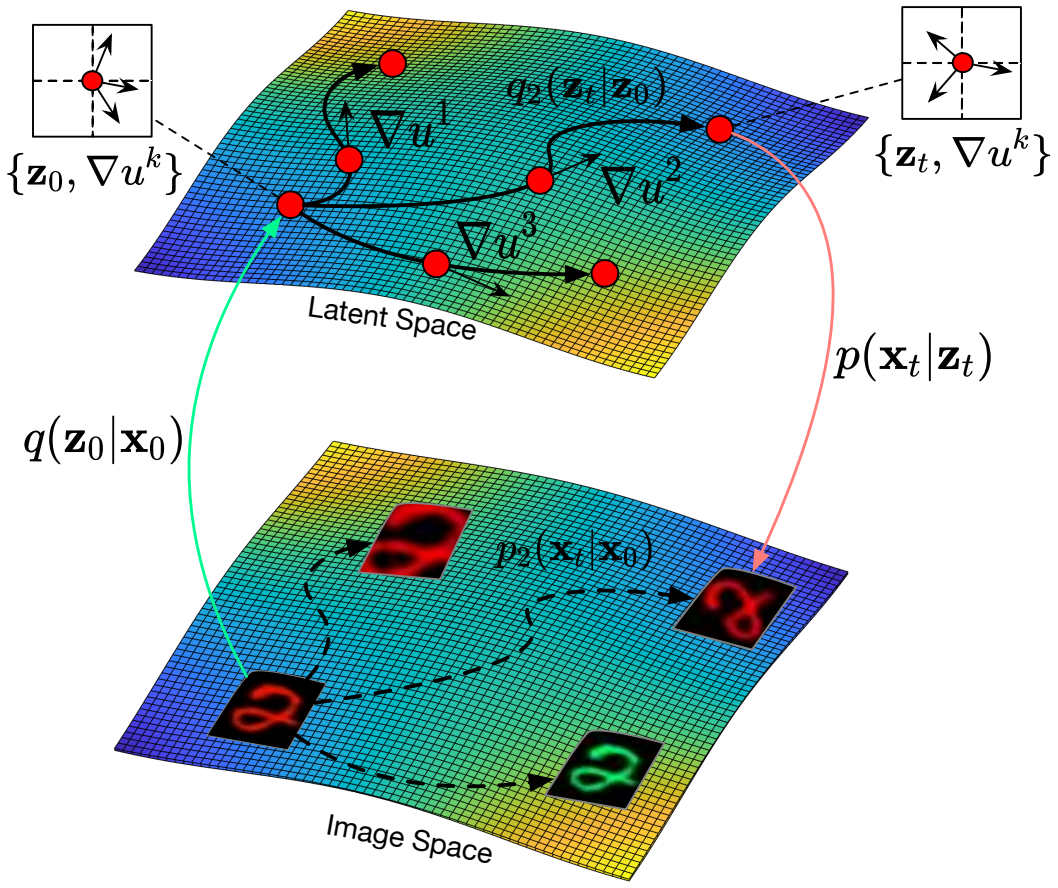


Figure 8.1: Illustration of our proposed flow factorized representation learning framework: at each point in the latent space we have a distinct set of tangent directions ∇u^k which define different transformations we would like to model in the image space. For each path, the latent sample evolves to the target on the potential landscape following dynamic optimal transport.

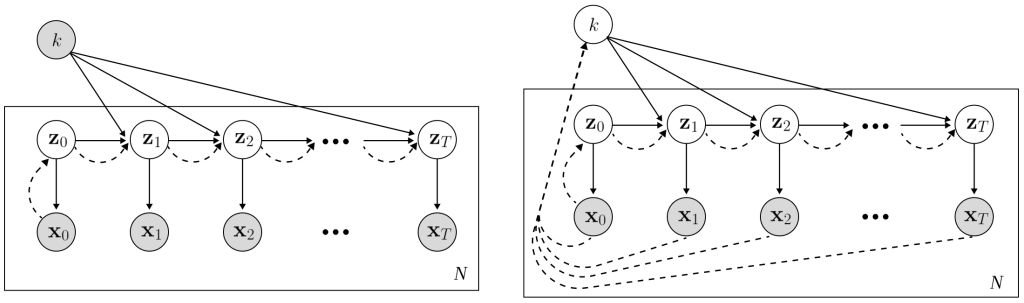


Figure 8.2: Depiction of our model in plate notation. (Left) Supervised, (Right) Weakly-supervised. White nodes denote latent variables, shaded nodes denote observed variables, solid lines denote the generative model, and dashed lines denote the approximate posterior. We see, as in a standard VAE framework, our model approximates the initial one-step posterior $p(\boldsymbol{z}_0|\boldsymbol{x}_0)$, but additionally approximates the conditional transition distribution $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)$ through dynamic optimal transport over a potential landscape.

In this section, we first introduce our generative model of sequences and then describe how we perform inference over the latent variables of this model in the next section.

Flow factorized sequence distributions

The model in this work defines a distribution over sequences of observed variables. We further factorize this distribution into k distinct components by assuming that each observed sequence is generated by one of the k separate flows of probability mass in latent space. Since in this work we model discrete sequences of observations $\bar{\boldsymbol{x}} = \{\boldsymbol{x}_0, \boldsymbol{x}_1 \ldots, \boldsymbol{x}_T\}$, we aim to define a joint distribution with a similarly discrete sequence of latent variables $\bar{\boldsymbol{z}} = \{\boldsymbol{z}_0, \boldsymbol{z}_1 \ldots, \boldsymbol{z}_T\}$, and a categorical random variable k describing the sequence type (observed or unobserved). Explicitly, we assert the following factorization of the joint distribution over T timesteps:

$$p(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, k) = p(k)p(\boldsymbol{z}_0)p(\boldsymbol{x}_0|\boldsymbol{z}_0)\prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)p(\boldsymbol{x}_t|\boldsymbol{z}_t). \tag{8.1}$$

Here $p(k)$ is a categorical distribution defining the transformation type, $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ asserts a mapping from latents to observations with Gaussian noise, and $p(\boldsymbol{z}_0) = \mathcal{N}(0, 1)$. A plate diagram of this model is depicted through the solid lines in Fig. 8.2.

Prior time evolution

To enforce that the time dynamics of the sequence define a proper flow of probability density, we compute the conditional update $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)$ from the continuous form of the continuity equation: $\partial_t p(\boldsymbol{z}) = -\nabla \cdot (p(\boldsymbol{z})\nabla\psi^k(\boldsymbol{z}))$, where $\psi^k(\boldsymbol{z})$ is the k'th prior potential function which advects the density $p(\boldsymbol{z})$ through the induced velocity field $\nabla\psi^k(\boldsymbol{z})$. Considering the discrete particle evolution corresponding to this density evolution, $\boldsymbol{z}_t = f(\boldsymbol{z}_{t-1}, k) = \boldsymbol{z}_{t-1} + \nabla_z\psi^k(\boldsymbol{z}_{t-1})$, we see that we can derive the conditional update from the continuous change of variables formula (D. Rezende and Mohamed, 2015; R. T. Q. Chen et al., 2018):

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k) = p(\boldsymbol{z}_{t-1})\left|\frac{df(\boldsymbol{z}_{t-1}, k)}{d\boldsymbol{z}_{t-1}}\right|^{-1} \tag{8.2}$$

In this setting, we see that the choice of ψ ultimately determines the prior on the transition probability in our model. As a minimally informative prior for random trajectories, we use a diffusion equation achieved by simply taking $\psi^k = -D_k \log p(\boldsymbol{z}_t)$. Then according to the continuity equation, the prior evolves as:

$$\partial_t p(\boldsymbol{z}_t) = -\nabla \cdot \Big(p(\boldsymbol{z}_t)\nabla\psi\Big) = D_k\nabla^2 p(\boldsymbol{z}_t) \tag{8.3}$$

where D_k is a constant coefficient that does not change over time. The density evolution of the prior distribution thus follows a constant diffusion process. We set D_k as a learnable parameter which is distinct for each k.

8.3. Flow factorized variational autoencoders

To perform inference over the unobserved variables in our model, we propose to use a variational approximation to the true posterior, and train the parameters of the model as a VAE. To do this, we parameterize an approximate posterior for $p(\boldsymbol{z}_0|\boldsymbol{x}_0)$, and additionally parameterize a set of K functions $u^k(\boldsymbol{z})$ to approximate the true latent potentials ψ^*. First, we will describe how we do this in the setting where the categorical random variable k is observed (which we call the supervised setting), then we will describe the model when k is also latent and thus additionally inferred (which we call the weakly supervised setting).

Inference with observed k (supervised)

When k is observed, we define our approximate posterior to factorize as follows:

$$q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k) = q(\boldsymbol{z}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k) \tag{8.4}$$

We see that, in effect, our approximate posterior only considers information from element $\boldsymbol{x}_0$; however, combined with supervision in the form of k, we find this is sufficient for the posterior to be able to accurately model full latent sequences. In the limitations section we discuss how the posterior could be changed to include all elements $\{\mathbf{x}_t\}_0^T$ in future work.

Combing Eq. (8.4) with Eq. (8.1), we can derive the following lower bound to the model evidence (ELBO):

$$\begin{aligned}\log p(\bar{\boldsymbol{x}}|k) &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log \frac{p(\bar{\boldsymbol{x}},\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\frac{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}{p(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\right] \\ &\geq \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log \frac{p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}},k)p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}},k)\right] + \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\left[\log \frac{p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\right]\end{aligned} \tag{8.5}$$

Substituting and simplifying, Eq. (8.5) can be re-written as

$$\log p(\bar{\boldsymbol{x}}|k) \geq \sum_{t=0}^{T} \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}}|k)}\big[\log p(\boldsymbol{x}_t|\boldsymbol{z}_t,k) - \mathrm{D}_{\mathrm{KL}}\left[q_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)||p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k)\right]\big] \tag{8.6}$$

We thus see that we have an objective very similar to that of a traditional VAE, except that our posterior and our prior now both have a time evolution defined by the conditional distributions.

Inference with latent k (weakly supervised)

When k is not observed, we can treat it as another latent variable, and simultaneously perform inference over it in addition to the sequence latents $\bar{\boldsymbol{z}}$. To achieve this, we define our approximate posterior and instead factorize it as

$$q(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}}) = q(k|\bar{\boldsymbol{x}})q(\boldsymbol{z}_0|\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1},k) \tag{8.7}$$

Following a similar procedure as in the supervised setting, we derive the new ELBO as

$$\begin{aligned}\log p(\bar{\boldsymbol{x}}) &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log \frac{p(\bar{\boldsymbol{x}},\bar{\boldsymbol{z}},k)}{q(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\frac{q(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}{p(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\right] \\ &\geq \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log \frac{p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}},k)p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\frac{p(k)}{q(k|\bar{\boldsymbol{x}})}\right] \\ &= \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log p(\bar{\boldsymbol{x}}|\bar{\boldsymbol{z}},k)\right] \\ &\quad + \mathbb{E}_{q_\theta(\bar{\boldsymbol{z}},k|\bar{\boldsymbol{x}})}\left[\log \frac{p(\bar{\boldsymbol{z}}|k)}{q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}},k)}\right] + \mathbb{E}_{q_\gamma(k|\bar{\boldsymbol{x}})}\left[\log \frac{p(k)}{q(k|\bar{\boldsymbol{x}})}\right]\end{aligned} \tag{8.8}$$

We see that, compared with Eq. (8.5), only one additional KL divergence term $\mathrm{D}_{\mathrm{KL}}\left[q_\gamma(k|\bar{\boldsymbol{x}})||p(k)\right]\big]$ is added. The prior $p(k)$ is set to follow a categorical distribution, and we apply the `Gumbel-SoftMax` trick (Jang et al., 2017) to allow for categorical re-parameterization and sampling of $q_\gamma(k|\bar{\boldsymbol{x}})$.

Posterior time evolution

As noted, to approximate the true generative model which has some unknown latent potentials ψ^k, we propose to parameterize a set of potentials as $u^k(\boldsymbol{z}, t) = \texttt{MLP}([\boldsymbol{z}; t])$ and train them through the ELBOs above. Again, we use the continuity equation to define the time evolution of the posterior, and thus we can derive the conditional time update $q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k)$ through the change of variables formula. Given the function of the sample evolution $\boldsymbol{z}_t = g(\boldsymbol{z}_{t-1}, k) = \boldsymbol{z}_{t-1} + \nabla_{\boldsymbol{z}} u^k$, we have:

$$q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k) = q(\boldsymbol{z}_{t-1})\left|\frac{dg(\boldsymbol{z}_{t-1}, k)}{d\boldsymbol{z}_{t-1}}\right|^{-1} \tag{8.9}$$

Converting the above continuous equation to the discrete setting and taking the logarithm of both sides gives the normalizing-flow-like density evolution of our posterior:

$$\log q(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, k) = \log q(\boldsymbol{z}_{t-1}) - \log|1 + \nabla^2_{\boldsymbol{z}} u^k| \tag{8.10}$$

The above relation can be equivalently derived from the continuity equation (*i.e.,* $\partial_t q(\boldsymbol{z}) = -\nabla\cdot\big(q(\boldsymbol{z})\nabla u^k\big)$). Notice that we only assume the initial posterior $q(\boldsymbol{z}_0|\boldsymbol{x}_0)$ follows a Gaussian distribution. For the future timesteps, we do not pose any further assumptions and just let the density evolve according to the sample motion.

Ensuring optimal transport of the posterior flow

As an inductive bias, we would like each latent posterior flow to follow the optimal transport path. To accomplish this, it is known that when the gradient ∇u^k satisfies certain PDEs, the evolution of the probability density can be seen to minimize the L_2 Wasserstein distance between the source distribution and the distribution of the target transformation. Specifically, we have:

Theorem 1 (Benamou-Brenier Formula (Benamou and Brenier, 2000))**.** *For probability measures μ_0 and μ_1, the L_2 Wasserstein distance can be defined as*

$$W_2(\mu_0, \mu_1)^2 = \min_{\rho, v}\left\{\int\int \frac{1}{2}\rho(x,t)|v(x,t)|^2\, dx\, dt\right\} \tag{8.11}$$

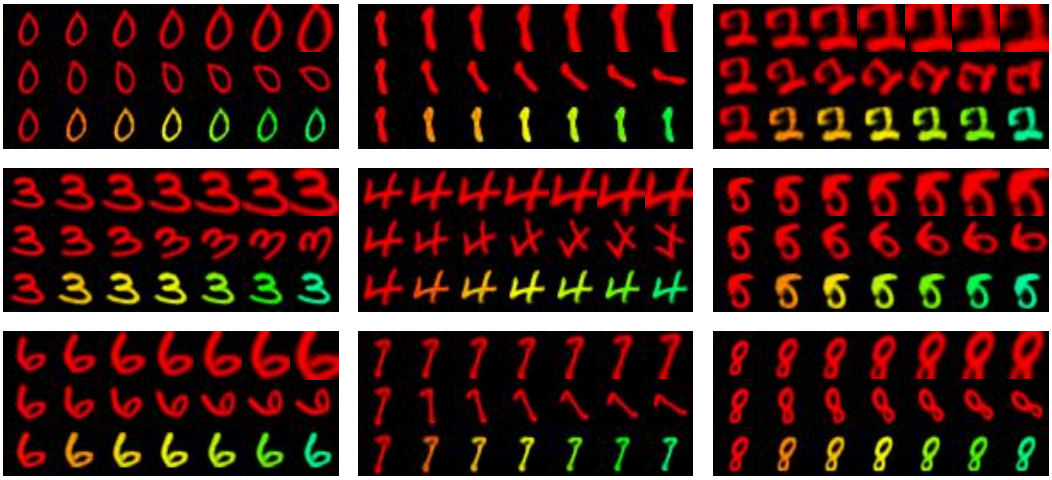


Figure 8.3: Exemplary latent evolution results of Scaling, Rotation, and Coloring on MNIST (LeCun, 1998). The top two rows are based on the supervised experiment, while the images of the bottom row are taken from the weakly-supervised setting of our experiment.

where the density ρ and the velocity v satisfy:

$$\frac{d\,\rho(x,t)}{dt} = -\nabla \cdot (v(x,t)\rho(x,t)),\ v(x,t) = \nabla u(x,t) \tag{8.12}$$

The optimality condition of the velocity is given by the generalized Hamilton-Jacobi (HJ) equation (*i.e.,* $\partial_t u + 1/2||\nabla u||^2 \leq 0$). The detailed derivation is deferred to the supplementary. We thus encourage our potential to satisfy the HJ equation with an external driving force as

$$\frac{\partial}{\partial t}u^k(z,t) + \frac{1}{2}||\nabla_z u^k(z,t)||^2 = f(z,t) \quad \text{subject to} \quad f(z,t) \leq 0 \tag{8.13}$$

Here we use another `MLP` to parameterize the external force $f(z,t)$ and realize the negativity constraint by setting $f(z,t) = -\texttt{MLP}([z;t])^2$. To achieve the PDE constraint, we impose a Physics-Informed Neural Network (PINN) (Raissi et al., 2019) loss as

$$\mathcal{L}_{HJ} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial}{\partial t}u^k(z,t) + \frac{1}{2}||\nabla_z u^k(z,t)||^2 - f(z,t)\right)^2 + ||\nabla u^k(z_0,0)||^2 \tag{8.14}$$

where the first term restricts the potential to obey the HJ equation, and the second term limits $u(z_t,t)$ to return no update at t=0, therefore matching the initial condition.

We note that an alternative formulation of this PINN constraint could be the traditional wave equation, as described in the introduction. In doing so, one would likely

see waves of activity traveling across the implicit cortical surface if measured. Indeed in prior work not included in this thesis (Y. Song, T. A. Keller, et al., 2023), we have experimented with such wave potentials for the purpose of latent traversals and found them to be highly effective. In this chapter, however, we leverage the connection between the HJ equation and dynamic optimal transport allowing us to introduce the valuable inductive bias of optimal transport into our framework. In ongoing future work, we are exploring ways to incorporate traveling wave dynamics more explicitly into this framework, allowing for a synergistic combination of optimal transport and traveling waves in deep latent variable models, and potentially giving another viewpoint from which to interpret traveling wave observations from neuroscience.

8.4. Experiments

This section starts with the experimental setup, followed by qualitative and quantitative results, and ends with discussions about the generalization ability to different composability and unseen data.

Datasets — We evaluate our method on two widely-used datasets in generative modeling, namely MNIST (LeCun, 1998) and Shapes3D (Burgess and H. Kim, 2018). For MNIST (LeCun, 1998), we manually construct three simple transformations including Scaling, Rotation, and Coloring. For Shapes3D (Burgess and H. Kim, 2018), we use the self-contained four transformations that consist of Floor Hue, Wall Hue, Object Hue, and Scale.

Baselines — We mainly compare our method with SlowVAE (D. Klindt et al., 2021) and Topographic VAE (TVAE) (T. A. Keller and Max Welling, 2021a) since these two baselines both achieve a form of approximate equivariance in a generative modeling framework. Specifically, as shown in Chapter 5, the TVAE can be seen to induce approximate equivariance through the latent 'Roll' operator, while SlowVAE enforces the Laplacian prior $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \prod \alpha\lambda/2\Gamma(1/\alpha) \exp\left(-\lambda|z_{t,i} - z_{t-1,i}|^{\alpha}\right)$ to sequential pairs. Within the disentanglement literature, our method is compared with the supervised PoFlow (Y. Song, T. A. Keller, et al., 2023) which adopts a wave-like potential flow for sample evolution, and the unsupervised β-VAE (Higgins, Matthey, et al., 2016) and FactorVAE (H. Kim and Mnih, 2018) which encourage independence between single latent dimensions. Finally, the vanilla VAE is used as a controlled baseline.

Metrics — We use the approximate equivariance error $\mathcal{E}_k$ and the log-likelihood of

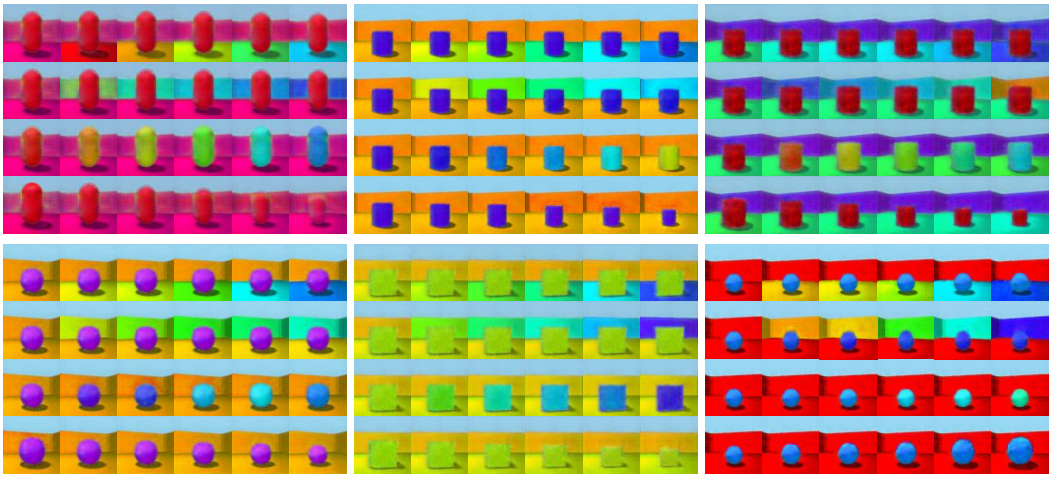


Figure 8.4: Exemplary latent flow results on Shapes3D (Burgess and H. Kim, 2018). The transformations from top to bottom are Floor Hue, Wall Hue, Object Hue, and Scale, respectively. The images of the top row are from the supervised experiment, while the bottom row is based on the weakly-supervised experiment.

transformed data $\log p(\boldsymbol{x}_t)$ as the evaluation protocols. The equivariance error is defined as $\mathcal{E}_k = \sum_{t=1}^{T} |\boldsymbol{x}_t - \texttt{Decode}(\boldsymbol{z}_t)|$ where $\boldsymbol{z}_t = \boldsymbol{z}_0 + \sum_{t=1}^{T} \nabla_{\boldsymbol{z}} u^k$. For TVAE, the latent operator is changed to $\texttt{Roll}(\boldsymbol{z}_0, t)$. For unsupervised disentanglement baselines (Higgins, Matthey, et al., 2016; H. Kim and Mnih, 2018) and SlowVAE (D. Klindt et al., 2021), we carefully select the latent dimension and tune the interpolation range to attain the traversal direction and range that correspond to the smallest equivariance error. Since the vanilla VAE does not have the corresponding learned transformation in the latent space, we simply set $\nabla_{\boldsymbol{z}} u^k = 0$ and take it as a lower-bound baseline. For all the methods, the results are reported based on 5 runs.

Notice that the above equivariance error is defined in the output space. Another reasonable evaluation metric is instead measuring error in the latent space as $\mathcal{E}_k = \sum_{t=1}^{T} |\texttt{Encode}(\boldsymbol{x}_t) - \boldsymbol{z}_t|$. We see the first evaluation method is more comprehensive as it further involves the decoder in the evaluation.

Methods	Supervision?	Equivariance Error (↓)			Log-likelihood (↑)
		Scaling	Rotation	Coloring	
VAE (Kingma and Max Welling, 2014)	No (✗)	1275.31±1.89	1310.72±2.19	1368.92±2.33	-2206.17±1.83
β-VAE (Higgins, Matthey, et al., 2016)	No (✗)	741.58±4.57	751.32±5.22	808.16±5.03	-2224.67±2.35
FactorVAE (H. Kim and Mnih, 2018)	No (✗)	659.71±4.89	632.44±5.76	662.18±5.26	-2209.33±2.47
SlowVAE (D. Klindt et al., 2021)	Weak (✓)	461.59±5.37	447.46±5.46	398.12±4.83	-2197.68±2.39
TVAE (T. A. Keller and Max Welling, 2021a)	Yes (✓)	505.19±2.77	493.28±3.37	451.25±2.76	-2181.13±1.87
PoFlow (Y. Song, T. A. Keller, et al., 2023)	Yes (✓)	234.78±2.91	231.42±2.98	240.57±2.58	-2145.03±2.01
Ours	Yes (✓)	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**	**-2112.45±1.57**
Ours	Weak (✓)	193.84±2.47	157.16±3.24	165.19±2.78	-2119.94±1.76

Table 8.1: Equivariance error $\mathcal{E}_k$ and log-likelihood $\log p(\boldsymbol{x}_t)$ on MNIST.

Qualitative results — Fig. 8.3 and 8.4 display decoded images of the latent evolution on MNIST (LeCun, 1998) and Shapes3D (Burgess and H. Kim, 2018), respectively. On both datasets, our latent flow can perform the target transformation precisely during evolution while leaving other traits of the image unaffected. In particular, for the weakly-supervised setting, the decoded images (*i.e.,* the bottom rows of Fig. 8.3 and 8.4) can still reproduce the given transformations well and it is even hard to visually tell them apart from the generated images under the supervised setting. This demonstrates the effectiveness of the weakly-supervised setting of our method, and implies that qualitatively our latent flow is able to learn the sequence transformations well under both supervised and weakly-supervised settings.

Methods	Supervision?	Equivariance Error (↓)				Log-likelihood (↑)
		Floor Hue	Wall Hue	Object Hue	Scale	
VAE (Kingma and Max Welling, 2014)	No (✗)	6924.63±8.92	7746.37±8.77	4383.54±9.26	2609.59±7.41	-11784.69±4.87
β-VAE (Higgins, Matthey, et al., 2016)	No (✗)	2243.95±12.48	2279.23±13.97	2188.73±12.61	2037.94±11.72	-11924.83±5.64
FactorVAE (H. Kim and Mnih, 2018)	No (✗)	1985.75±13.26	1876.41±11.93	1902.83±12.27	1657.32±11.05	-11802.17±5.69
SlowVAE (D. Klindt et al., 2021)	Weak ( )	1247.36±12.49	1314.86±11.41	1102.28±12.17	1058.74±10.96	-11674.89±5.74
TVAE (T. A. Keller and Max Welling, 2021a)	Yes (✓)	1225.47±9.82	1246.32±9.54	1261.79±9.86	1142.01±9.37	-11475.48±5.18
PoFlow (Y. Song, T. A. Keller, et al., 2023)	Yes (✓)	885.46±10.37	916.71±10.49	912.48±9.86	924.39±10.05	-11335.84±4.95
Ours	Yes (✓)	**613.29±8.93**	**653.45±9.48**	**605.79±8.63**	**599.71±9.34**	**-11215.42±5.71**
Ours	Weak ( )	690.84±9.57	717.74±10.65	681.59±9.02	653.58±9.57	-11279.61±5.89

Table 8.2: Equivariance error $\mathcal{E}_k$ and log-likelihood $\log p(\boldsymbol{x}_t)$ on Shapes3D.

Quantitative results — Tables 8.1 and 8.2 compare the equivariance error and the log-likelihood on MNIST (LeCun, 1998) and Shapes3D (Burgess and H. Kim, 2018), respectively. Our method learns the latent flows which model the transformations precisely, achieving the best performance across datasets under different supervision settings. Specifically, our method outperforms the previous best baseline by 69.74 on average in the equivariance error and by 32.58 in the log-likelihood on MNIST. The performance gain is also consistent on Shapes3D: our method surpasses the second-best baseline by 291.70 in the average equivariance error and by 120.42 in the log-likelihood. In the weakly-supervised setting, our method also achieves very competitive performance, falling behind that of the supervised setting in the average equivariance error slightly by 6.22 on MNIST and by 67.88 on Shapes3D.

8.5. Discussion

Extrapolation: switching transformations — In Fig. 8.5 we demonstrate that, empowered by our method, it is possible to switch latent transformation categories mid-way through the latent evolution and maintain coherence. That is, we perform $\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \nabla_{\boldsymbol{z}} u^k$ for $t \leq T/2$ and then change to $\boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \nabla_{\boldsymbol{z}} u^j$ where $j \neq k$ for $t > T/2$. As can be seen, the factor of variation immediately changes after the trans-

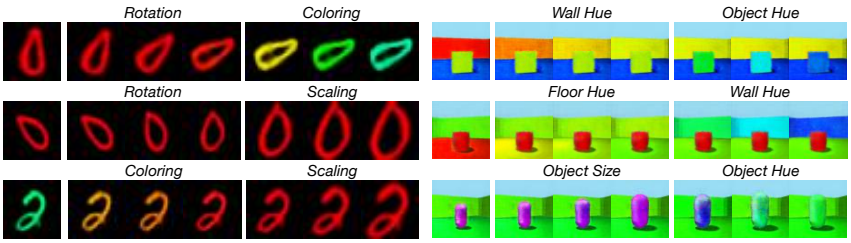


Figure 8.5: Exemplary visualization of switching transformations during the latent sample evolution.

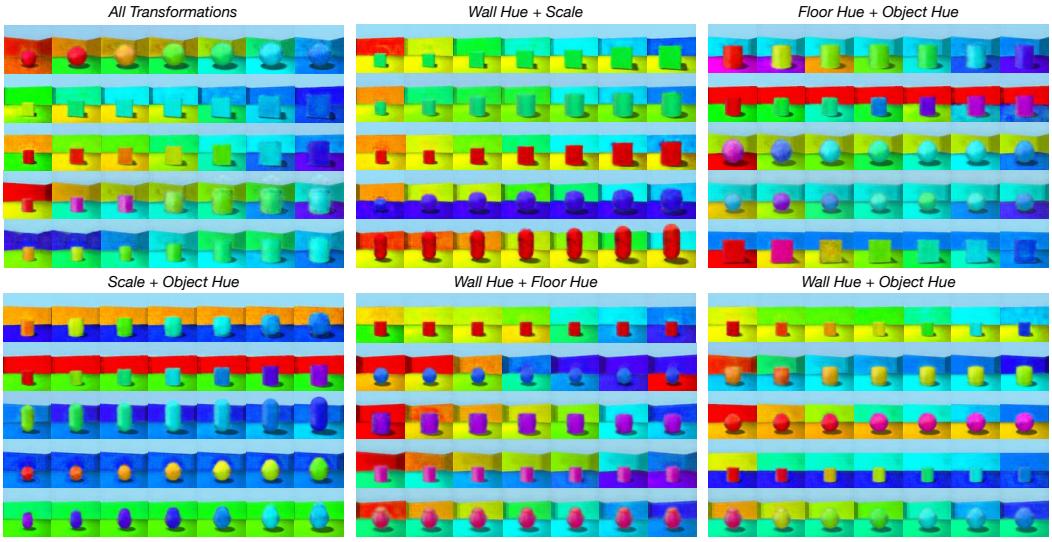


Figure 8.6: Examples of combining different transformations simultaneously during the latent evolution.

formation type is switched. Moreover, the transition phase is smooth and no other attributes of the image are influenced.

Extrapolation: superimposing transformations — Besides switching transformations, our method also supports applying different transformations simultaneously, *i.e.,* consistently performing $z_t = z_{t-1} + \sum_k^K \nabla_z u^k$ during the latent flow process. Fig. 8.6 presents such exemplary visualizations of superimposing two, and all, transformations simultaneously. In each case, the latent evolution corresponds to simultaneous smooth variations of multiple image attributes. This indicates that our method also generalizes well to superposing different transformations.

Notice that we only apply single and separate transformations in the training stage. Switching or superposing transformations in the test phase can be thus understood as an extrapolation test to measure the generalization ability of the learned equivariance to novel compositions.

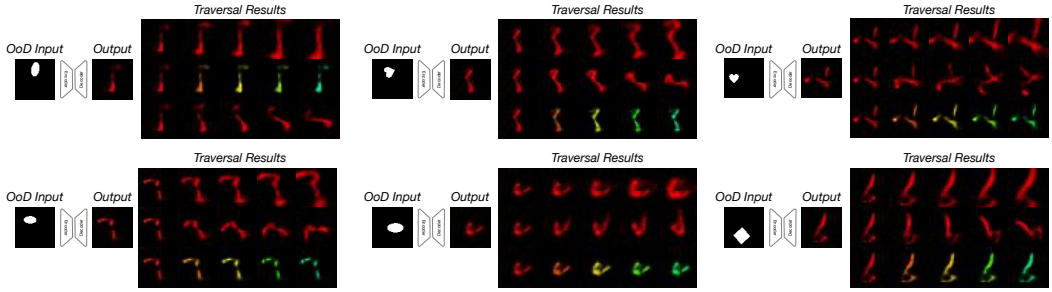


Figure 8.7: Equivariance generalization to unseen OoD input data. Here the model is trained on MNIST but the latent flow is tested on dSprites.

Equivariance generalization to new data — We also test whether the learned equivariance holds for Out-of-Distribution (OoD) data. To verify this, we validate our method on a test dataset that is different from the training set and therefore unseen to the model. Fig. 8.7 displays the exemplary visualization results of the VAE trained on MNIST but evaluated on dSprites (Matthey et al., 2017). Although the reconstruction quality is poor, the learned equivariance is still clearly effective as each transformation still operates as expected: scaling, rotation, and coloring transformations from top to bottom respectively.

8.6. Related work

Disentangled representation learningx — The idea of learning disentangled representation dates back to factorizing non-redundant input patterns (Schmidhuber, 1992) but is recently first studied by InfoGAN (Xi Chen et al., 2016) and β-VAE (Higgins, Matthey, et al., 2016). InfoGAN (Xi Chen et al., 2016) achieves disentanglement by maximizing the mutual information between a subset of latent dimensions and observations, while β-VAE (Higgins, Matthey, et al., 2016) induces the factorized posterior $q(\mathbf{z})$ by penalizing the Total Correlation (TC) through an extra hyperparameter $\beta>1$ controlling the strength of the KL divergence. Following infoGAN, many attempts have been made to facilitate the discovery of semantically meaningful traversal directions through regularization (Goetschalckx et al., 2019; Jahanian et al., 2020; Voynov and Babenko, 2020; Härkönen et al., 2020; X. Zhu et al., 2020; Peebles et al., 2020; Shen and Zhou, 2021; Wei et al., 2021; J. Zhu, Feng, et al., 2021; Tzelepis et al., 2021; J. Zhu, Shen, et al., 2022; Y. Song, Sebe, et al., 2022; Oldfield et al., 2023). The follow-up research of β-VAE mainly explored different methods to factorize the aggregated posterior (Dilokthanakul et al., 2016; Dupont, 2018; Kumar et al., 2018; H. Kim and Mnih, 2018; R. T. Chen et al., 2018; Y. Jeong and H. O. Song, 2019; Ding et al., 2020; Locatello et al., 2020; Tai et al., 2022).

More recently, some works proposed to disentangle diffusion models by discovering meaningful directions in the bottleneck of denoising networks (Kwon et al., 2023; Park et al., 2023; T. Yang et al., 2023). The previous literature mainly considers disentanglement as learning different transformations per dimension or per linear direction. Our method generalizes this concept to learning a distinct tangent bundle ∇u^k that moves every latent sample via dynamic OT.

We see the most similar method to ours is the work of (Y. Song, T. A. Keller, et al., 2023). In (Y. Song, T. A. Keller, et al., 2023), the authors also apply the gradient of a potential function to move the latent code. However, their potentials are restricted to obey the wave equations, which do not really correspond to the OT theory. Also, they do not consider the posterior evolution but instead use the loss $||\boldsymbol{z}_t - \texttt{Encode}(\boldsymbol{x}_t)||^2$ to match the latent codes. By contrast, we propose a unified probabilistic generative model that encompasses the posterior flow that follows dynamic OT, the flow-like time evolution, and different supervision settings.

Equivariant neural networks — Equivariance has been considered a desired inductive bias for deep neural networks as this property can preserve geometric symmetries of the input space (Geoffrey E Hinton, Krizhevsky, et al., 2011a; Schmidt and Roth, 2012; C.-Y. Lee et al., 2015; Lenc and Vedaldi, 2015; Agrawal et al., 2015). Analytically equivariant networks typically enforce explicit symmetry to group transformations in neural networks (T. Cohen and Max Welling, 2016a; Taco S Cohen and Max Welling, 2017; Ravanbakhsh et al., 2017; D. E. Worrall et al., 2017; D. Worrall and Max Welling, 2019b; Van der Pol et al., 2020; Finzi, Stanton, et al., 2020; Hoogeboom et al., 2022). Another line of research proposed to directly learn approximate equivariance from data (Diaconu and D. Worrall, 2019b; Connor et al., 2021; D. Klindt et al., 2021; Dey et al., 2021; T. A. Keller and Max Welling, 2021a). Our framework re-defines approximate equivariance by matching the latent probabilistic flow to the actual path of the given transformation in the image space.

Optimal transport in deep learning — There is a vast literature on OT theory and applications in various fields (Villani, 2009; Villani, 2021). Here we mainly highlight the relevant applications in deep learning. The pioneering work of (Cuturi, 2013) proposed a light-speed implementation of the Sinkhorn algorithm for fast computation of entropy-regularized Wasserstein distance, which opened the way for many differentiable Sinkhorn algorithm-based applications (Frogner et al., 2015; Feydy et al., 2019; Chizat et al., 2020; Eisenberger et al., 2022; Kolouri et al., 2021). In generative modeling, the Wasserstein distance is often used to minimize

the discrepancy between the data distribution and the model distribution (Arjovsky, Chintala, et al., 2017; Tolstikhin et al., 2018; Salimans et al., 2018; Patrini et al., 2020). Inspired by the fluid mechanical interpretation of OT (Benamou and Brenier, 2000), some normalizing flow methods (D. Rezende and Mohamed, 2015; Dinh et al., 2017; Kingma and Dhariwal, 2018) considered regularizing the velocity fields to satisfy the HJ equation, thus matching the dynamic OT plan (L. Yang and Karniadakis, 2020; Finlay et al., 2020; Tong et al., 2020; Onken et al., 2021; Neklyudov et al., 2023). Our method applies PINNs (Raissi et al., 2019) to directly model generalized HJ equations in the latent space and uses the gradient fields of learned potentials to generate latent flows, which also aligns to the theory of dynamic fluid mechanical OT.

8.7. Conclusion

In this chapter, we introduce Flow Factorized Representation Learning which defines a set of latent flow paths that correspond to sequences of different input transformations. The latent evolution is generated by the gradient flow of learned potentials following dynamic optimal transport. Our setup re-interprets the concepts of both *disentanglement* and *equivariance*. Extensive experiments demonstrate that our model achieves higher likelihoods on standard representation learning benchmarks while simultaneously achieving smaller equivariance error. Furthermore, we show that the learned latent transformations generalize well, allowing for flexible composition and extrapolation to new data.

8.8. Limitations

For flexibility and efficiency, we use PINN (Raissi et al., 2019) constraints to model the HJ equation. However, such PDE constraints are approximate and not strictly enforced. Other PDE modeling approaches include accurate neural PDE solvers (Hsieh et al., 2019; Brandstetter et al., 2022; Richter-Powell et al., 2022) or other improved PINN variants such as competitive PINNs (Zeng et al., 2023) and robust PINNs (Bajaj et al., 2023). Also, when infering with observed k, we change the posterior from $q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k)$ to $q(\bar{\boldsymbol{z}}|\boldsymbol{x}_0, k)$ because we assume k contains sufficient information of the whole sequence. To keep the posterior definition of $q(\bar{\boldsymbol{z}}|\bar{\boldsymbol{x}}, k)$, we need to make $q(\boldsymbol{z}_t)$ also a function of $\boldsymbol{x}_t$. This can be achieved either by changing the potential to $u(\boldsymbol{z}_{t-1}, \boldsymbol{x}_t, t-1)$ or modifying the external driving force to $f(\boldsymbol{z}_{t-1}, \boldsymbol{x}_t, t-1)$. Nonetheless, we see these modifications would make the model

less flexible than our current formulations as the element $\boldsymbol{x}_t$ might be needed during inference.

Part III

Structure-based Learning

Chapter 9

SPATIO-TEMPORAL STRUCTURE AS SELF-SUPERVISION

9.1. Introduction

As described in the introduction to this thesis, an inflection point in the history of modern deep learning is known as the 'ImageNet Moment' where deep neural networks were shown to dramatically surpass the image-classification performance of existing computer vision techniques for the first time. Why did this eye-opening event happen at precisely this time? What precisely were the obstacles that were overcome and the preconditions that were met to enable such a perfect storm? Most importantly, what can studying these past obstacles tell us about obstacles that may be hindering modern deep learning?

One element that undoubtedly contributed to the success of this moment was the growing understanding of the best practices of training deep neural networks, and specifically those pertaining to convolutional neural networks. A major component of these best practices was the training of such architectures efficiently on modern graphics processing units (GPUs). This yielded a training speed increase which suddenly made training of large scale (state-of-the-art competitive) networks feasible.

However, going hand-in-hand with this scaling-up of network size was the scaling-up of the associated training set size. In hindsight, it can now be seen that perhaps this may have been an equally important precondition for this eye-opening event to happen when it did. Specifically, the dataset from which this moment inherits its name, the ImageNet dataset, contains over 1,000 unique classes each with 1,000 images of the respective objects in each class. At the time of its introduction this was the largest labeled dataset of its kind in the computer vision community, and with over 1 million human-annotated images it required a significant organizational effort to gather and process (Deng et al., 2009). As we described in the introduction, deep neural networks are known to be analogous to bottomless pits in the context of data requirements. By suddenly having such a massive amount of organized information at their fingertips, Alex, Ilya, and Geoff were able to finally satisfy the data-hunger of these models and demonstrate their potential.

From the perspective of cognitive scientists, psychologists, and neuroscientists how-

ever, this success was bittersweet. It was clear that humans did not require this much labeled data, and therefore these systems must not be representative of the learning processes of naturally intelligent systems. Furthermore, from the perspective of the rapidly developing start-up industry around image classification, this necessity for massive amounts of labeled data for every unique image classification task made practical deployment of these models significantly more challenging than one would initially hope for.

As a counteraction to the clear limitation of massive data requirements, scientists began to work on algorithms which required little to no supervised labels, but instead tried to solve what were called 'auxiliary tasks' to learn useful representations of the data. This training paradigm came to be known as self-supervised learning (SSL), since the networks could be seen to be extracting their own form of supervision directly from the data itself (LeCun and Misra, 2021). Although many of these self-supervised learning algorithms have since been shown to be highly related in theory and concept to the unsupervised learning algorithms of the previous chapters, in practice they have shown to work better for learning representations of data which are useful for downstream tasks such as classification (at least in the context of today's paradigms).

In light of these successes, many of the prominent proponents of self-supervised learning, such as Yann LeCun, have argued that they believe that natural intelligence is likely performing some type of self-supervised learning in order to avoid the unrealistic data requirements we see with standard supervised learning (LeCun and Misra, 2021). However, there are still many known discrepancies with how these algorithms actually perform learning and how we believe the brain learns. Foremost, the vast majority of these algorithms still rely the backpropagation algorithm to communicate the extracted supervision signal back from the output of the network to each of the intermediate neurons. As we described in Section 3.3 it is precisely the computational machinery required for this exact credit assignment which makes the backpropagation algorithm biologically implausible given our current understanding of neurobiology and its mapping to the abstractions of artificial neural networks (Lillicrap, Santoro, et al., 2020). Beyond biological considerations, backpropagation also has heavy compute and memory requirements which make it less than ideal for practical reasons as well. For example the amount of memory required to run the algorithm scales with the number of layers (or time steps in recurrent networks) as all steps must be stored in memory in order to compute the gradient.

As a result of these differences in learning and the ensuing practical limitations, the study of novel biologically plausible learning algorithms has become an active area of research for scientists sharing the goal of this thesis: to close the observed gaps between natural and artificial intelligence. One promising step in that direction came in 2019 with Löwe, O'Connor, et al. (2019) demonstrating that deep neural networks could be trained in a block-wise fashion with existing self-supervised learning algorithms, thereby eliminating the need for 'end-to-end' training of such systems. However, in the ensuring years, the popular self-supervised learning algorithms have diverged from compatibility with this approach, turning instead to what is known as 'augmentation-based' self-supervised learning which makes such a layer-wise extension significantly more challenging. In this chapter, we will present our own contribution on this front. Specifically, we will make a direct comparison between the augmentations used in modern self-supervised learning and the structure in natural and artificial neural networks that we have discussed up to this point. In doing so, we will provide a blueprint for how we believe such structure may be beneficial for self-supervised learning itself, and thereby breathe new life into the local self-supervised learning ideas of Löwe, O'Connor, et al. (2019).

In detail, to accomplish this, in the following sections we will study self-supervised learning algorithms when equivariant neural networks are used as the main 'backbone' feature extractors. Interestingly, in this setting we will find a theoretical convergence of existing loss functions from the literature, and ultimately generalize these with the framework of *Homomorphic Self-Supervised Learning*. Experimentally, we will show that, when the assumption of an augmentation-homomorphic backbone is satisfied, this framework subsumes input augmentation, as evidenced by identical performance over a range of settings. We further validate this theory by showing that when our assumption is not satisfied, the framework fails to learn useful representations. Finally, we explore the new generalized parameters introduced by this framework, demonstrating an immediate path forward for improvements to existing SSL methods which operate without input augmentations.

9.2. Background

In this section, we will provide quick review of the relevant aspects of group-equivariant neural networks that will be used in this chapter. We will then review general self-supervised frameworks and how prior literature differs with respect to its use of input augmentations.

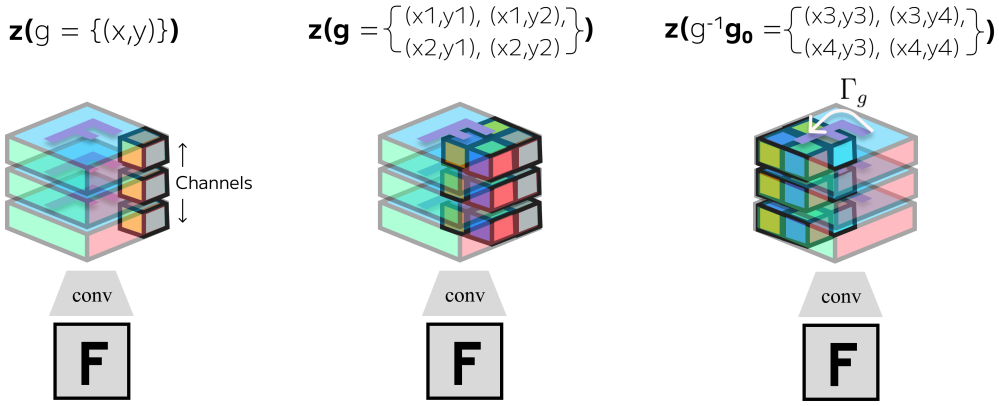


Figure 9.1: Visualization of a 'fiber' (left), a 'fiber bundle' (center) and a group representation Γ_g acting on a fiber bundle (right). We see that a fiber is defined as all features at an individual group element (in this case all feature channels at an individual spatial dimension), while a fiber bundle is all features at a set of ordered group elements. In this figure, we depict feature channels stacked along the z-dimension, different from the 'lifted' dimension in Figure 9.2 (left).

Equivariance

Formally, a map which preserves the structure of the input space in the output space is termed a homomorphism. The most prominent example of a homomorphism in modern deep learning is the class of group equivariant neural networks, which are analytically constrained to be group homomorphisms for specified transformation groups (such as translation, rotation, mirroring, and scaling). The map $f : \mathcal{X} \to \mathcal{Z}$ is said to be equivariant with respect to the group $\mathcal{G} = (G, \cdot)$ if

$$\exists \Gamma_g \quad \text{such that} \quad f(\tau_g[\boldsymbol{x}]) = \Gamma_g[f(\boldsymbol{x})] \quad \forall g \in G \,, \tag{9.1}$$

where G is the set of all group elements, $\cdot$ is the group operation, τ_g is the representation of the transformation $g \in G$ in input space $\mathcal{X}$, and Γ_g is the representation of the same transformation in output space $\mathcal{Z}$. If τ_g and Γ_g are formal group representations (Serre, 1977) such maps f are termed group-homomorphisms since they preserve the structure of the group representation τ_g in input space with the output representation Γ_g. There are many different methods for constructing group equivariant neural networks, resulting in different representations of the transformation in feature space Γ_g. In this work, we consider only discrete groups $\mathcal{G}$ and networks which admit regular representations for Γ.

Group-Convolutional Neural Networks — One common way in which group-equivariant networks are constructed is via the group-convolution (G-conv) (T. Cohen and Max Welling, 2016b) as described in Section 3.2. In this chapter, we will consider dis-

crete groups $\mathcal{G}$, and we will denote the pre-activation output of a $\mathcal{G}$-equivariant convolutional layer l as $\boldsymbol{z}^l$, with a corresponding input $\boldsymbol{y}^l$. In practice these values are stored in finite arrays with a feature multiplicity equal to the order of the group in each space. Explicitly, $\boldsymbol{z}^l \in \mathbb{R}^{C_{out} \times |G_{out}|}$, and $\boldsymbol{y}^l \in \mathbb{R}^{C_{in} \times |G_{in}|}$ where G_{out} and G_{in} are the set of group elements in the output and input spaces respectively. We use the following shorthand for indexing $\boldsymbol{z}^l(g) \equiv \boldsymbol{z}^{l,:,g} \in \mathbb{R}^{C_{out}}$ and $\boldsymbol{y}^l(g) \equiv \boldsymbol{y}^{l,:,g} \in \mathbb{R}^{C_{in}}$, denoting the vector of feature channels at a specific group element (sometimes called a 'fiber' (T. Cohen and M. Welling, 2017)). Explicitly then, the value $z^{l,c}(g) \in \mathbb{R}$ of a single output at layer l, channel c and element g is

$$z^{l,c}(g) \equiv [\boldsymbol{y}^l \star \boldsymbol{\psi}^{l,c}](g) = \sum_{h \in G_{in}} \sum_{i}^{C_{in}} y^{l,i}(h) \psi_i^{l,c}(g^{-1} \cdot h) \,, \tag{9.2}$$

where $\boldsymbol{\psi}_i^{l,c}$ is the filter between the i^{th} input channel (subscript) and the c^{th} output channel (superscript), and is similarly defined (and indexed) over the set of input group elements G_{in}. We note that this equation differs from Equation 3.8 only in the fact that this definition now includes a sum over input channels C_{in} and an extra index c to denote output channels C_{out}.

The representation Γ_g can then be defined as $\Gamma_g[\boldsymbol{z}^l(h)] = \boldsymbol{z}^l(g^{-1} \cdot h)$ for all $l > 0$ when $\mathcal{G}_{in}^l = \mathcal{G}_{out}^l = \mathcal{G}_{out}^0$. We see that Γ_g is a 'regular representation' of the group, meaning that it acts by permuting features along the group dimension while leaving feature channels intact. Group equivariant layers can then be composed with pointwise non-linearities and biases to yield a fully equivariant deep neural network (e.g. $\boldsymbol{y}_i^{l+1} = \text{ReLU}(\boldsymbol{z}^l + \boldsymbol{b})$ where $\boldsymbol{b} \in \mathbb{R}^{C_{out}}$ is a learned bias shared over the output group dimensions). For $l = 0$, $\boldsymbol{y}^0$ is set to the raw input $\boldsymbol{x}$, and typically the input group is set to the group of all 2D integer translations up to height H and width W: $\mathcal{G}_{in}^0 = (\mathbb{Z}_{HW}^2, +)$. The output group $\mathcal{G}_{out}^0$ is then chosen by the practitioner and is typically a larger group which includes translation as a subgroup, e.g. the roto-translation group, or the group of scaling & translations. In this way, the first layer of a group-equivariant neural network is frequently called the 'lifting layer' since it lifts the input from the translation group, containing only spatial dimensions, to a larger group by adding an additional 'lifted' dimension.

Example — As a simple example, a standard convolutional layer would have all height (H) and width (W) spatial coordinates as the set G_{out}, giving $\boldsymbol{z} \in \mathbb{R}^{C \times HW}$. A group-equivariant neural network (T. Cohen and Max Welling, 2016b) which is equivariant with respect to the the group of all integer translations and 90-degree

rotations ($p4$) would thus have a feature multiplicity four times larger ($\boldsymbol{z} \in \mathbb{R}^{C\times 4HW}$), since each spatial element is associated with the four distinct rotation elements ($0^o, 90^o, 180^o, 270^o$). Such a rotation equivariant network is depicted in Figure 9.2 with the 'lifted' rotation dimension extended along the vertical axis (θ). In both the translation and rotation cases, the regular representation Γ_g acts by permuting the representation along the group dimension, leaving the feature channels unchanged.

Notation — In the remainder of this paper we will see that it is helpful to have a notation which allows for easy reference to the sets of features corresponding to multiple group elements simultaneously. These sets are sometimes called 'fiber bundles' and are visually compared with individual fibers in Figure 9.1. In words, a fiber (left) can be described as all features values at a specific group element (such as all channels at a given spatial location), and a fiber bundle (center) is then all features at an ordered set of group elements (such as all channels for a given spatial patch). We denote the set of fibers corresponding to an ordered set of group elements $\boldsymbol{g}$ as: $\boldsymbol{z}(\boldsymbol{g}) = [\boldsymbol{z}(g) \mid g \in \boldsymbol{g}] \in \mathbb{R}^{|\boldsymbol{g}|C_{out}}$. Using this notation, we can define the action of Γ_g as: $\Gamma_g[\boldsymbol{z}(\boldsymbol{g}_0)] = \boldsymbol{z}(g^{-1} \cdot \boldsymbol{g}_0)$. Thus Γ_g can be seen to move the fibers from 'base' locations $\boldsymbol{g}_0$ to a new ordered set of locations $g^{-1}\cdot\boldsymbol{g}_0$, as depicted in on the right side of Figure 9.1. We highlight that order is critical for our definition since a transformation such as rotation may simply permute $\boldsymbol{g}_0$ while leaving the unordered set intact.

Self-Supervised Learning

As mentioned in the introduction, self-supervised learning can be seen as extracting a supervision signal from the data itself, often by means of transformations applied to the input. Many terms in self-supervised learning objectives can thus often be abstractly written as a function $\mathcal{I}(\boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)})$ of two batches of vectors $\boldsymbol{V}^{(1)} = \{\mathbf{v}_i^{(1)}\}_{i=1}^N$ and $\boldsymbol{V}^{(2)} = \{\mathbf{v}_i^{(2)}\}_{i=1}^N$ where there is some relevant relation between the elements of the two batches. In this description, we see that there are two main degrees of freedom which we will explore in the following paragraphs: the choice of function $\mathcal{I}$, and the precise relationship between $\boldsymbol{V}^{(1)}$ and $\boldsymbol{V}^{(2)}$.

SSL Loss Functions: $\mathcal{I}^C$ and $\mathcal{I}^{NC}$ — The most prominent SSL loss terms in the literature are often segregated into contrastive $\mathcal{I}^{\mathrm{C}}$ (T. Chen et al., 2020; Oord et al., 2018) and non-contrastive $\mathcal{I}^{\mathrm{NC}}$ (Grill et al., 2020; Xinlei Chen and He, 2020) losses. At a high level, *contrastive losses* frequently rely on a vector similarity function $\mathrm{sim}(\cdot,\cdot)$ (such as cosine similarity), and 'contrast' its output for 'positive' and 'negative' pairs. A general form of a contrastive loss, inspired by the 'InfoNCE'

loss (Oord et al., 2018), can be written as:

$$\mathcal{I}_i^{\mathrm{C}}(\boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)}) = -\frac{1}{N}\log\frac{\exp\Big(\mathrm{sim}\big(h(\mathbf{v}_i^{(1)}), h(\mathbf{v}_i^{(2)})\big)/T\Big)}{\sum_{j\neq i}^{N}\sum_{k,l}^{2}\exp\Big(\mathrm{sim}\big(h(\mathbf{v}_i^{(k)}), h(\mathbf{v}_j^{(l)})\big)/T\Big)} \tag{9.3}$$

where h is a non-linear 'projection head' $h : \mathcal{Z} \to \mathcal{Y}$ and T is the 'temperature' of the softmax. We see that such losses can intuitively be thought of as trying to classify the correct 'positive' pair (given by $\mathbf{v}_i^{(1)}$ & $\mathbf{v}_i^{(2)}$) out of a set of negative pairs (given by all other pairs in the batch). Comparatively, *non-contrastive losses* are often applied to the same sets of representations $\boldsymbol{V}^{(1)}$ and $\boldsymbol{V}^{(2)}$, but crucially forego the need for 'negative pairs' through other means of regularization (such as a stop-gradient on one branch (Xinlei Chen and He, 2020; Yuandong Tian, Xinlei Chen, et al., 2021) observed to regularize the eigenvalues of the representation covariance matrix). Fundamentally this often results in a loss of the form:

$$\mathcal{I}_i^{\mathrm{NC}}(\boldsymbol{V}^{(1)}, \boldsymbol{V}^{(2)}) = -\frac{1}{N}\mathrm{sim}\big(h(\mathbf{v}_i^{(1)}), \mathrm{SG}(\mathbf{v}_i^{(2)})\big)\,, \tag{9.4}$$

where SG denotes the stop-gradient operation. In this work we focus the majority of our experiments on the $\mathcal{I}^{\mathrm{NCE}}$ loss specifically. However, given this general formulation which decouples the specific loss from the choice of pairs $\boldsymbol{V}^{(1)}$ & $\boldsymbol{V}^{(2)}$, and the fact that our framework only operates on the formulation of the pairs, we will see that our analyses and conclusions extend to all methods which can be written this way. In the following, we will introduce the second degree of freedom which captures many SSL algorithms: the precise relationship between $\boldsymbol{V}^{(1)}$ and $\boldsymbol{V}^{(2)}$.

Relationship Between SSL Pairs: $\boldsymbol{V}^{(1)}$ *&* $\boldsymbol{V}^{(2)}$ — Similar to our treatment of SSL loss functions $\mathcal{I}$, in this section we separate the existing literature into two categories with respect to the leveraged relationship between positives pairs. Specifically, we compare methods which rely on input augmentations, which we call Augmentation-based SSL (A-SSL), to methods which operate entirely within the representation of a single input, which we call Feature-space SSL (F-SSL). An influential framework which relies on augmentation is the SimCLR framework (T. Chen et al., 2020). Using the above notation, this is given as:

$$\mathcal{L}_i^{\text{A-SSL}}(\mathbf{X}) = \mathop{\mathbb{E}}_{g_1,g_2\sim G}\mathcal{I}_i^{\mathrm{C}}\left(\left\{f\big(\tau_{g_1}[\boldsymbol{x}_n]\big)\right\}_n^N, \left\{f\big(\tau_{g_2}[\boldsymbol{x}_n]\big)\right\}_n^N\right), \tag{9.5}$$

where $\tau_g[\boldsymbol{x}]$ denotes the action of the sampled augmentation g on the input, G is the set of all augmentations, and $f(\boldsymbol{x}) = \mathbf{v}$ is the backbone feature extractor to be trained.

This loss is then summed over all elements i in the batch before backpropagation. In this work, we consider this SimCLR loss given in Equation 9.5 as the canonical A-SSL method given its broad adoption and similarity with other augmentation-based methods. The second class of SSL methods we consider in this work are those which operate without the use of explicit input augmentations, but instead compare subsets of a representation for a single image directly. Models such as Deep InfoMax (DIM(L)) (Hjelm et al., 2019), Greedy InfoMax (GIM) (Löwe, O'Connor, et al., 2019), and Contrastive Predictive Coding (CPC) (Oord et al., 2018)[1] can all be seen to be instantiations of such Feature-space SSL methods. At a low level, these methods vary in the specific subsets of the representations which are used in the loss (from single spatial elements to larger 'patches'), and vary in the similarity function (with some using a log-bilinear model $\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(\boldsymbol{a}^T W \boldsymbol{b}\right)$, instead of cosine similarity). In this work we define a general Feature-space SSL (F-SSL) loss in the spirit of these models which similarly operates in the feature space of a single image, uses an arbitrary spatial 'patch' size $|\boldsymbol{g}|$, and a cosine similarity function. Formally:

$$\mathcal{L}_i^{\text{F-SSL}}(\mathbf{X}) = \mathop{\mathbb{E}}_{\boldsymbol{g}_1, \boldsymbol{g}_2 \sim \mathbb{Z}^2_{HW}} \mathcal{I}_i^{\text{C}}\left(\left\{\boldsymbol{z}_n(\boldsymbol{g}_1)\right\}_n^N, \left\{\boldsymbol{z}_n(\boldsymbol{g}_2)\right\}_n^N\right), \tag{9.6}$$

where $\boldsymbol{g} \sim \mathbb{Z}^2_{HW}$ refers to sampling a contiguous patch from the spatial coordinates of a convolutional feature map, and $\boldsymbol{z}_n$ is the output of our backbone $f(\boldsymbol{x}_n)$. In the following section, we show how equivariant backbones unify these two losses into a single loss, helping to explain both their successes and limitations while additionally demonstrating clear directions for their generalization.

9.3. Homomorphic Self-Supervised Learning

In this section we introduce Homomorphic Self-Supervised Learning (H-SSL) as a general framework for SSL with homomorphic encoders, and further show it both generalizes and unifies many existing SSL algorithms.

To begin, consider an A-SSL objective such as Equation 9.5 when f is equivariant with respect to the input augmentation. By the definition of equivariant maps in Equation 9.1, the augmentation commutes with the feature extractor: $f(\tau_g[\boldsymbol{x}]) = \Gamma_g[f(\boldsymbol{x})]$. Thus, replacing $f(\boldsymbol{x}_n)$ with its output $\boldsymbol{z}_n = \boldsymbol{z}_n(\boldsymbol{g}_0)$, and applying the

[1] In CPC, the authors use an autoregressive encoder to encode one element of the positive pairs. In GIM, they find that in the visual domain, this autoregressive encoder is not necessary, and thus the loss reduces to simple contrasting the representations from raw patches with one another, as defined here.

definition of the operator, we get:

$$\mathcal{L}_i^{\text{H-SSL}}(\mathbf{X}) = \mathop{\mathbb{E}}_{g_1, g_2 \sim G} \mathcal{I}_i^{\text{C}}\left(\left\{\boldsymbol{z}_n\left(g_1^{-1} \cdot \boldsymbol{g}_0\right)\right\}_n^N, \left\{\boldsymbol{z}_n\left(g_2^{-1} \cdot \boldsymbol{g}_0\right)\right\}_n^N\right). \tag{9.7}$$

Ultimately, we see that $\mathcal{L}^{\text{H-SSL}}$ subsumes the use of input augmentations by defining the 'positive pairs' as two fiber bundles from *the same representation* $\boldsymbol{z}_n$, simply indexed using two differently transformed base spaces $g_1^{-1} \cdot \boldsymbol{g}_0$ and $g_2^{-1} \cdot \boldsymbol{g}_0$ (depicted in Figure 9.2, and Figure 9.1, center & right). Interestingly, this loss highlights the base space $\boldsymbol{g}_0$ as a parameter choice previously unexplored in the A-SSL frameworks. In Section 9.4 we empirically explore different choices of $\boldsymbol{g}_0$ and comment on their consequences.

A second interesting consequence of this derivation is the striking similarity of the $\mathcal{L}^{\text{H-SSL}}$ objective and other existing SSL objectives which operate without explicit input augmentations to generate multiple views. This can be seen most simply by comparing $\mathcal{L}^{\text{H-SSL}}$ from Equation 9.7 with the $\mathcal{L}^{\text{F-SSL}}$ objective from Equation 9.6. Specifically, since $\boldsymbol{g}_1$ & $\boldsymbol{g}_2$ from the F-SSL loss can be decomposed as a single base patch $\boldsymbol{g}_0$ offset by two single translation elements g_1 & g_2 (e.g. $\boldsymbol{g}_1 = g_1^{-1}\boldsymbol{g}_0$ and $\boldsymbol{g}_2 = g_2^{-1}\boldsymbol{g}_0$), we see that Equation 9.6 can be derived directly from Equation 9.7 by setting $G = \mathbb{Z}_{HW}^2$ and the size of the base patch $|\boldsymbol{g}_0|$ equal to the size of the patches used for each F-SSL case. Consequently, these F-SSL losses are contained in our framework where the set of 'augmentations' ($\mathcal{G}$) is the 2D translation group, and the base space ($\boldsymbol{g}_0$) is a small subset of the spatial coordinates. Since $\mathcal{L}^{\text{H-SSL}}$ is also derived directly from $\mathcal{L}^{\text{A-SSL}}$ (when f is equivariant), we see that it provides a means to unify these previously distinct sets of SSL objectives. In Section 9.4 we validate this theoretical equivalence empirically. Furthermore, since $\mathcal{L}^{\text{H-SSL}}$ is defined for transformation groups beyond translation, it can be seen to generalize F-SSL objectives in a way that we have not previously seen exploited in the literature. In Section 9.4 we include a preliminary exploration of this generalization to scale and rotation groups.

9.4. Experiments

In this section, we empirically validate the derived equivalence of A-SSL and H-SSL in practice, and further reinforce our stated assumptions by demonstrating how H-SSL objectives (and by extension F-SSL objectives) are ineffective when representational structure is removed. We study how the parameters of H-SSL (topographic distance) relate to those traditionally used in A-SSL (augmentation

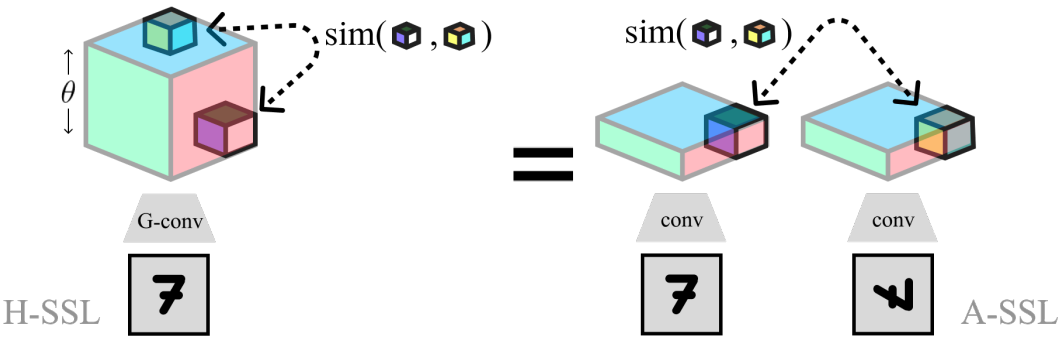


Figure 9.2: Overview of Homomorphic-SSL (left) and its relation to traditional Augmentation-based SSL (right). Positive pairs extracted from the lifted dimension (θ) of a rotation equivariant network (G-conv) are equivalent to pairs extracted from the separate representations of two rotated images.

strength), and finally explore how the new parameter generalizations afforded by our framework (such as choices of $\boldsymbol{g}_0$ and $\mathcal{G}$) impact performance.

Empirical Validation

For perfectly equivariant networks f, and sets of transformations which exactly satisfy the group axioms, the equivalence between Equations 9.5 and 9.7 is exact. However, in practice, due to aliasing, boundary effects, and sampling artifacts, even for simple transformations such as translation, equivariance has been shown to not be strictly satisfied (R. Zhang, 2019). In Table 9.1 we empirically validate our proposed theoretical equivalence between $\mathcal{L}^{\text{A-SSL}}$ and $\mathcal{L}^{\text{H-SSL}}$, showing a tight correspondence between the downstream accuracy of linear classifiers trained on representations learned via the two frameworks.

Precisely, for each transformation (Rotation, Translation, Scale), we use a backbone network which is equivariant specifically with respect to that transformation (e.g. rotation equivariant CNNs, regular CNNs, and Scale Equivariant Steerable Networks (SESN) (Sosnovik et al., 2020)). For A-SSL we augment the input at the pixel level by: randomly translating the image by up to ± 20% of its height/width (for translation), randomly rotating the image by one of $[0^o, 90^o, 180^o, 270^o]$ (for rotation), or randomly downscaling the image to a value between 0.57 & 1.0 of its original scale. These two augmented versions of the image are then fed through the backbone separately, and a single fiber (meaning $|\boldsymbol{g}_0| = 1$) is randomly selected. We investigate the impact of the base space size separately in Section 9.4. For H-SSL we use no in-

put augmentations and instead rely on differently indexed base patches (formed by shifting the randomly selected fiber $\boldsymbol{g}_0$ by two separate randomly selected group elements g_1 & g_2). For example, for A-SSL with translation, we compare the feature vectors for two translated images *at the same pixel location* $\boldsymbol{g}_0$. For H-SSL with translation, we compare the feature vectors of a single image at two *translated locations* $g_1^{-1} \cdot \boldsymbol{g}_0$ & $g_2^{-1} \cdot \boldsymbol{g}_0$. Ultimately, we see an equivalence between the performance of the A-SSL models and H-SSL models which significantly differs from the frozen and supervised baselines, validating our theoretical conclusions from Section 9.3.

Table 9.1: MNIST (LeCun and Cortes, 2010), CIFAR10 (Krizhevsky, Nair, et al., n.d.) and Tiny ImageNet (TIN) (Y. Le and X. S. Yang, 2015) top-1 test accuracy (mean ± std. over 3 runs) of a detached classifier trained on the representations from SSL methods with different backbones. We compare A-SSL and H-SSL with random frozen and fully supervised backbones. We see equivalence between A-SSL and H-SSL from the first two columns, as desired, and often see a significant improvement in performance for H-SSL methods when moving from Translation to generalized groups such as Scale.

Dataset	Transformation	Backbone	A-SSL	H-SSL	Frozen	Supervised
MNIST	Rotation	Rot-Eq.	68.2 ± 2.5	70.3 ± 5.4	*87.2 ± 0.8*	*99.4 ± 0.1*
	Translation	CNN	95.9 ± 0.3	96.0 ± 1.3	*94.1 ± 0.3*	*99.2 ± 0.1*
	Scale	SESN	98.6 ± 0.1	98.3 ± 0.2	*94.7 ± 0.6*	*99.3 ± 0.1*
CIFAR10	Rotation	Rot-Eq.	46.1 ± 0.6	48.3 ± 0.5	*38.4 ± 0.1*	*73.0 ± 1.1*
	Translation	CNN	39.2 ± 0.5	36.3 ± 1.1	*40.4 ± 0.2*	*76.2 ± 1.4*
	Scale	SESN	59.4 ± 0.2	56.7 ± 0.4	*41.1 ± 0.6*	*78.0 ± 0.2*
TIN	Rotation	Rot-Eq.	14.9 ± 0.3	13.5 ± 0.5	*6.1 ± 0.2*	*22.5 ± 0.1*
	Scale	SESN	16.2 ± 0.4	14.0 ± 1.3	*6.4 ± 0.2*	*23.7 ± 0.2*

H-SSL Without Structure

To further validate our assertion that $\mathcal{L}^{\text{H-SSL}}$ requires a homomorphism, in Table 9.2 we show the same models from Table 9.1 without equivariant backbones. Explicitly, we use the same overall model architectures but replace the individual layers with non-equivariant counterparts. Specifically, for the MLP, we replace the convolutional layers with fully connected layers (slightly reducing the total number of activations from 6272 to 2048 to reduce memory consumption), and replace the SESN kernels of the scale-equivariant models with fully-parameterized, non-equivariant counterparts, otherwise keeping the output dimensionality the same (resulting in the

$6\times$ larger output dimension). Furthermore, for these un-structured representations, in the H-SSL setting, we 'emulate' a group dimension to sample 'fibers' from. For the MLP we do this by reshaping the 2048 dimensional output to (16,128), and select one of the 16 rows at each iteration. For the CNN, we similarly use the 6 times larger feature space to sample $\frac{1}{6}^{th}$ of the elements as if they were scale-equivariant.

We thus observe that when equivariance is removed, but all else remains equal, $\mathcal{L}^{\text{H-SSL}}$ models perform significantly below their input-augmentation counterparts, and similarly to a 'frozen' randomly initialized backbone baselines, indicating the learning algorithm is no longer effective. Importantly, this indicates why existing F-SSL losses (such as DIM(L) (Hjelm et al., 2019)) always act within equivariant dimensions (e.g. between the spatial dimensions of feature map pixels) – these losses are simply ineffective otherwise. An intuitive understanding of this result can be given by viewing arbitrary features as being related by some unknown input transformation which may not preserve the target information about the input. In contrast, however, since equivariant dimensions rely on symmetry transforms, contrast over such dimensions is known to be equivalent to contrasting transformed inputs.

Table 9.2: An extension of Table 9.1 with non-equivariant backbones. In each setting, the backbone is set to have a number of outputs equivalent to the equivariant counterpart, allowing for us to compute the H-SSL objective identically to before. We see that the H-SSL methods perform similar to, or worse than, the frozen baseline when equivariance is removed, as expected.

Dataset	Transformation	Backbone	A-SSL	H-SSL	Frozen	Supervised
MNIST	Translation	MLP	87.6 ± 0.2	58.2 ± 0.5	83.0 ± 0.8	98.6 ± 0.1
	Scale	CNN	95.2 ± 0.1	87.2 ± 2.4	87.2 ± 0.6	99.3 ± 0.1
CIFAR10	Scale	CNN	53.6 ± 0.2	37.5 ± 0.1	43.6 ± 0.3	67.9 ± 2.1

Parameters of H-SSL

Base size $|\mathbf{g}_0|$ — As discussed in Section 9.3, The H-SSL framework identifies new parameter choices such as the base space $\mathbf{g}_0$. This parameter specifically carries special importance since it is the main distinction between the A-SSL and F-SSL losses in the literature. Specifically, the size of $\mathbf{g}_0$ is set to the full representation size in the SimCLR framework, while it is typically a small patch or an individual

pixel in F-SSL losses such as DIM(L) or GIM. To investigate the impact of this difference, we explore the performance of the H-SSL framework as we gradually increase the size of $\mathbf{g}_0$ from 1 (akin to DIM(L) losses) to $|G| - 1$ (akin to SimCLR), with no padding. In each setting, we similarly increase the dimensionality of the input layer for the non-linear projection head h to match the multiplicative increase in the dimension of the input representation $\boldsymbol{z}(\mathbf{g})$. In Figure 9.3 (left) we plot the %-change in top-1 accuracy on CIFAR-10 for each size. We see a minor increase in performance as we increase the size, but note relative stability, again suggesting greater unity between A-SSL and H-SSL.

Topographic Distance — Each augmentation in a standard SimCLR augmentation stack is typically associated with a scalar or vector valued 'strength'. For example, this can correspond to the maximum number of pixels translated, the range of rescaling, or the maximum number of degrees to be rotated. We note that the same concept is present in the H-SSL framework and is defined by the associated latent representation of the transformation. For networks which use regular representations (as in this work), the degree of a transformation corresponds exactly to the degree of shift within the representation. We thus propose that an analogous concept to augmentation strength is *topographic distance* in equivariant networks, meaning the distance between the two sampled fiber bundles as computed along the group dimensions (i.e. the 'degree of shift'). For example, for convolution, this would correspond to the number of feature map pixels between two patches. For scale, this would correspond to the number of scales between two patches. In Figure 9.3 (right), we explore how the traditional notion of augmentation 'strength' can be equated with the 'topographic distance' between g_1 and g_2 and their associated fibers (with a fixed base size of $|\mathbf{g}_0| = 1$). Here we approximate topographic distance as the maximum euclidean distance between sampled group elements for simplicity ($||g_1 - g_2||_2^2$), where a more correct measure would be computed using the topology of the group. We see, in alignment with prior work (Yonglong Tian, Sun, et al., 2020; Yonglong Tian, Krishnan, et al., 2019), that the strength of augmentation (and specifically translation distance) is an important parameter for effective self supervised learning, likely relating to the mutual information between fibers as a function of distance.

Methods

Model Architectures — All models presented in this paper are built using the convolutional layers from the SESN (Sosnovik et al., 2020) library for consistency

and comparability. For scale equivariant models, we used the set of 6 scales $[1.0, 1.25, 1.33, 1.5, 1.66, 1.75]$. To construct the rotation equivariant backbones, we use only a single scale of $[1.0]$ and augment the basis set with four 90-degree rotated copies of the basis functions at $[0^o, 90^o, 180^o, 270^o]$. These rotated copies thus defined the group dimension. This technique of basis or filter-augmentation for implementing equivariance is known from prior work and has been shown to be equivalent to other methods of constructing group-equivariant neural networks (B. Li et al., 2021). For translation models, we perform no basis-augmentation, and again define the set of scales used in the basis to be a single scale $[1.0]$, thereby leaving only the spatial coordinates of the final feature maps to define the output group. On MNIST (LeCun and Cortes, 2010), we used a backbone network f composed of three SESN convolutional layers with 128 final output channels, ReLU activations and BatchNorms between layers. The output of the final ReLU is then considered our z for contrastive learning (for $\mathcal{L}^{\text{A-SSL}}$ and $\mathcal{L}^{\text{H-SSL}}$) and is of shape $(128, S \times R, 8, 8)$ where S is the number of scales for the experiment (either 1 or 6), and R is the number of rotation angles (either 1 or 4). On CIFAR10 and Tiny ImageNet we used SESN-modified ResNet18 and ResNet20 models respectively where the output of the last ResNet blocks were taken as z for contrastive learning. For all models where translation is not the studied transformation, we average pool over the spatial dimensions to preserve consistent input-dimensionality to the nonlinear projection head.

Training Details — For training, we use the LARS optimizer (You et al., 2017) with an initial learning rate of 0.1, and a batch size of 4096 for all models. We use an NCE temperature (T) of 0.1, half-precision training, a learning rate warm-up of 10 epochs, a cosine lr-update schedule, and weight decay of 1×10^{-4}. On MNIST we train for 500 epochs and on CIFAR10 and Tiny ImagNet (TIN) we train for 1300 epochs.

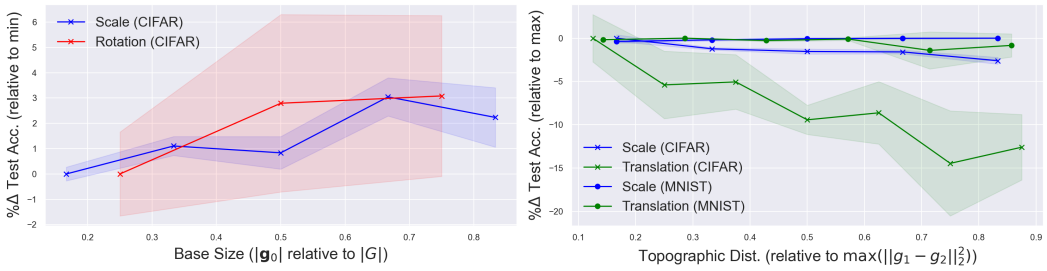


Figure 9.3: Study of the impact of new H-SSL parameters on top-1 test accuracy. (Left) Test accuracy marginally increases as we increase total base space size g_0. (Right) Test accuracy is constant or decreases as we increase the maximum distance between fiber bundles considered positive pairs.

Our work is built upon the literature from the fields equivariant deep learning and self-supervised learning as outlined in Sections 9.1 and 9.2. Beyond this background, our work is highly related in motivation to a number of studies specifically related to equivariance in self-supervised learning.

Undesired Invariance in SSL — One subset of recent prior work has focused on the undesired invariances learned by A-SSL methods (Xiao et al., 2021; Tsai, T. Li, et al., 2022) and on developing methods by which to avoid this through learned approximate equivariance (Dangovski et al., 2022; Wang, Geng, et al., 2021). Our work is, to the best of our knowledge, the first to suggest and validate that the primary reason for the success of feature-space SSL objectives such as DIM(L) (Hjelm et al., 2019) and GIM (Löwe, O'Connor, et al., 2019) is due to their exploitation of (translation) equivariant backbones (i.e. CNNs). Furthermore, while prior work shows benefits to existing augmentation-based SSL objectives when equivariance is induced, our work investigates how equivariant representations can directly be leveraged to formulate new theoretically-grounded SSL objectives. In this way, these two approaches may be complimentary.

Data Augmentation in Feature Space — There exist multiple works which can similarly be interpreted as performing data augmentation in feature space both for supervised and self-supervised learning. These include Dropout (Srivastava et al., 2014), Manifold Mixup (Verma et al., 2018), and others which perform augmentation directly in feature space (DeVries and Taylor, 2017; Hendrycks et al., 2020), or through generative models (Sandfort et al., 2019). We see that our work is fundamentally different from these in that it is not limited to simply performing an augmentation which would have been performed in the input in latent space. Instead, it maximally leverages structured representations to generalize all of these approaches and show how others can be included under this umbrella. Specifically, a framework such as DIM(L) is not explicitly performing an augmentation in latent space, but rather comparing two subsets of a representation which are offset by an augmentation. As we discuss in Section 9.6, this distinction is valuable for developing novel SSL algorithms which can substitute learned homomorphisms for learned augmentations – potentially sidestepping challenges associated with working in input-space directly.

Hybrid A-SSL + F-SSL — Some recent work can be seen to leverage both augmentation-based and feature-space losses simultaneously. Specifically, Augmented Multiview

Deep InfoMax (Bachman et al., 2019) achieves exactly this goal and is demonstrated to yield improved performance over its non-hybrid counterparts. Although similar in motivation, and perhaps performance, to our proposed framework, the Homomorphic SSL framework differs by unifying the two losses into a single objective, rather than a sum of two separate objectives.

9.6. Discussion

In this chapter we have studied the impact of combining augmentation-homomorphic feature extractors with augmentation-based SSL objectives. In doing so, we have introduced a new framework we call Homomorphic-SSL which illustrates an equivalence between previously distinct SSL methods when the homomorphism constraint is satisfied. Using this framework, we demonstrated that when the constraint is not satisfied, feature-space based SSL methods fail to learn valuable representations, shedding some light on why existing F-SSL methods do succeed. Furthermore, we investigated the new parameters highlighted by our model, such as the base size $|\boldsymbol{g}_0|$ and the transformation group $\mathcal{G}$ for F-SSL, showing unexploited potential for improvement of existing methods by tuning of these parameters.

We present this work as an attempt to renew interest in SSL objectives which operate without multiple inferences of a transformed image, such as Deep InfoMax (Hjelm et al., 2019) and Greedy InfoMax (Löwe, O'Connor, et al., 2019), by allowing them to exploit the theoretical foundations developed for multi-view SSL (Tosh et al., 2020; Yuandong Tian, Yu, et al., 2020; Tsai, Wu, et al., 2020; Kügelgen et al., 2021; Federici et al., 2020). Although F-SSL methods have to-date not yielded the same performance as their A-SSL counterparts, we believe the coupling between objective and network architecture is likely to yield more parallelizable algorithms which are therefore more scalable and biologically plausible, as has been demonstrated in prior work (Löwe, O'Connor, et al., 2019). In this way, such algorithmic advances could additionally yield potential insights into how biological neural networks could perform a type of self-supervised learning.

Limitations — Despite the unification of existing methods, and benefits from generalization, we note that this approach is still limited. Specifically, the equivalence between $\mathcal{L}^{\text{A-SSL}}$ and $\mathcal{L}^{\text{H-SSL}}$, and the benefits afforded by this equivalence, can only be realized if it is possible to analytically construct a neural network which is equivariant with respect to the transformations of interest. Since it is not currently known how to construct neural networks which are analytically equivariant with respect to

all input augmentations used in modern SSL, this constraint is precisely the greatest current limitation of this framework. Although the field of equivariant deep learning has made significant progress in recent years, state of the art techniques are still restricted to E(n) and continuous compact and connected Lie Groups (Finzi, Stanton, et al., 2020; Finzi, Max Welling, et al., 2021; Cesa et al., 2022; Weiler and Cesa, 2019). We believe in this regard, our analysis sheds some light on the success of methods which perform data augmentation over those which operate directly in feature space in recent literature – it is simply too challenging with current methods to construct models with structured representations for the diversity of transformations needed to induce a sufficient set of invariances for linear separability of classes. We therefore propose this work not as an immediate improvement to the state of the art, but rather as a new perspective on SSL which provides a bridge to previously distant literature.

Future Work — In light of this, we believe that our framework specifically suggests a novel path forward via learned homomorphisms, (T. A. Keller and Max Welling, 2021a; Keurti et al., 2022; Connor et al., 2021; Dehmamy et al., 2021; Pal and Savvides, 2018). In the H-SSL framework, a learned homomorphism can be seen as equivalent to a learned augmentation, providing a potential new avenue for approaching the extremely challenging (Blaas et al., 2021) but fruitful (Y. Shi et al., 2022) goal of learned image augmentations.

Conclusion

Chapter 10

CONCLUSION

The unfolding story of artificial intelligence has long been a dance with the insights gained from theoretical and experimental neuroscience. As told throughout this thesis, this narrative has been mapped out by our ever increasing understanding of natural neural structure, and spans from Alan Turing's unorganized machines to Kunihiko Fukushima's Neocognitron.

In the last decades, our ability to measure neural activity has increased at a rate likely unimaginable to the early pioneers who initiated this dance nearly a century ago. With the development of functional magnetic resonance imaging (fMRI) we gained the ability to map localized neural selectivity across the entirety of the human cortex simultaneously and efficiently. With the development of multi-electrode arrays such as the Neuropixel we increased our single neuron recording abilities by orders of magnitude, allowing for simultaneous recording of hundreds of individual neurons over time spans of more than two months (Steinmetz et al., 2021). As anecdotally reported by some experimental neuroscientists, the amount of data we are able to record with such devices in a single day would have taken nearly a year of work just less than a decade ago, if it would have been possible at all.

With this newfound wealth of information, what insights might we be able to distill if we look closely? Might we be able to discover the novel inductive biases which will define the next generation of artificial neural networks, just as the inductive biases of McCulloch and Pitts' artificial neurons have defined modern artificial intelligence?

In this thesis, we have focused on a number of these new insights from the neuroscience community that were previously unknown to the early architects of artificial intelligence. Specifically, we have discussed structure in natural neural networks, built models to emulate this structure, and studied the ensuing relationships between such models and more canonical forms of structure in machine learning. We have further leveraged these models to test hypotheses from theoretical and computational neuroscience, providing empirical support for some while simultaneously introducing novel methods to the machine learning community in the contexts of both long-term memory and self-supervised learning. In conclusion, we return to the research questions we outlined in our motivation, and see how far we have come

towards answering them.

10.1. Research Question 1: Spatial Structure

In the first part of this thesis, we aimed to explore if we may be able to devise a computational explanation for the role of topographic organization in the brain. Specifically, we asked: *What role does topographic organization play in the computational functions of the brain?*

In Chapter 4, we provide an argument for how the principle of redundancy reduction may be a factor behind the widespread observation of topographic organization throughout the cortex. To reiterate and summarize that argument here, *according to the framework of topographic generative modeling, topographic organization arises as an attempt to accurately model higher order correlations between latent variables*. If one were to ignore such higher order correlations, and instead use the common assumption of independant latent variables (as in ICA), the prior can be interpreted as mis-specified for many natural datasets. For example, natural image datasets are known to contain such correlations when decomposed in terms of Gabor or wavelet filters (Lyu and E. P. Simoncelli, 2009a). Topographic organization then comes from the fact that, in order to construct a prior distribution which has such higher order correlations (what we call a topographic prior), a simple method is to use a hierarchical generative model which pools over sets of latent variables to induce higher order correlation (i.e. the construction of our $\mathbf{T}$ variable in Section 4.4: $\mathbf{T} = \frac{\mathbf{Z}}{\sqrt{\mathbf{W}\mathbf{U}^2}}$).

At first glance, it may appear that our model is primarily motivated by information theoretic concerns, and therefore our results imply the answer to our first research question is that topographic organization is an outcome of the brain being an optimized information processing system. While we do believe that this is part of the story, there is one more part which relies on a key assumption not stated in this above argument: the assumption that the latent variables which are pooled together are pooled *locally*. In our work, we accomplish this through the $\mathbf{W}$ matrix which implements local average pooling. The question we must ask ourselves then is, why must this be the case? For example, the lower level neurons (latent variables $\mathbf{U}$) could be shuffled all over the cortex, and as long as the appropriate sets are pooled together, a prior with the correct higher order correlation statistics will be formed – the variable $\mathbf{T}$ would be none-the-wiser, simply shuffled itself and therefore not containing any local topographic organization. The answer appears to come down to one of the well known theories of topographic organization: wiring-length minimization (Koulakov

and Chklovskii, 2001; Essen, 1997). If lower level neurons were shuffled all over the cortex, the appropriate pooling operation would necessarily have to extend over greater distances than if those neurons were located more closely to one-another.

Ultimately then, the model in our work appears to suggest a two-fold answer to the question of topographic organization's computational role. To put it simply, the idea of wiring length minimization argues that neurons which must frequently communicate should be located nearby one-another to reduce distance dependant costs, and the principle of redundancy reduction gives an argument for why neurons with similar selectivity necessarily must communicate with one another – to accurately model higher order correlations. Combined, these ideas yield a generative modeling explanation of topographic organization which our work shows can be used to produce localized category selectivity similar to that of higher level visual cortices.

We highlight that while this answer is appealing from a Bayesian perspective of brain modeling, this is simply the conclusion that we believe should be drawn from the results presented in this thesis, and not a conclusion which should override other theories. There are many competing theories for the emergence and role of topographic organization and our results are far from ruling these theories out. If anything, we believe that, as we have seen in the above analysis, by combining insights from multiple theories we are likely to arrive at a stronger conclusion than any in isolation. As further evidence of this, in the following subsection we will overview how equivariance may be a second computational role for topographic organization, as described in Chapter 5.

10.2. Research Questions 2 & 3: Spatio-Temporal Structure

In the second part of this thesis, we aimed to explore if we may be able to perform a similar type of analysis to understand the more dynamic forms of spatial structure in neural representations, namely spatio-temporal strucutre. Specifically, we asked two complimentary questions: *Does spatio-temporal structure play a role in the computational functions of the brain? & Can natural spatio-temporal structure be efficiently and beneficially implemented in deep neural network architectures?*

In Chapters 5, 6, 7 & 8 we provided four separate (but related) methods by which spatio-temporal structure could be efficiently implemented in deep neural network architectures. In doing so, we showed that models with spatio-temporal structure were (i) better able to model data sets with extensive symmetries, (ii) able to structure

their latent space in a usable manner to allow for controlled generation, and (iii) better able to maintain memories over longer time spans.

These results clearly demonstrate that the answer to question 3 is unarguably affirmative – spatio-temporal structure is typically simple to implement (sometimes only requiring a single convolution operation) and can serve multiple independant functions.

With respect to question 2, our results appear to imply that the answer may be again affirmative; however, as with question 1, we urge caution in over interpreting our results. In the above chapters we have successfully demonstrated that in some abstract settings, spatio-temporal structure reminiscent of that observed in natural neural networks is beneficial for achieving task-relevant goals. However, it is still unknown if these computational purposes are precisely the same as those which the brain leverages these structures for. Our results provide empirical evidence for hypotheses from neuroscience, however we believe it is only neuroscience itself which can truly answer these questions, at least with the current state of our computational models.

10.3. Research Question 4: Supervision from Structure

In the final part of our thesis, we explored whether the neural representational structure studied throughout this work may be beneficial for developing a learning algorithm itself. Specifically, we asked: *Can spatio-temporal representational structure be leveraged to perform efficient and local learning without labeled data?*

In Chapter 9 we showed how structured representations may serve as an alternative for data augmentation in the framework of self-supervised learning, allowing for the alleviation of many of the biologically implausible aspects of modern SSL algorithms. Specifically, we argued that through a combination of our proposed homomorphic self-supervised learning and the ideas of Greedy Infomax (Löwe, O'Connor, et al., 2019), one could construct modern self-supervised learning algorithms which avoid the need for end-to-end backpropagation and data augmentation while maintaining high performance. Such an algorithm could be seen as a form of structured predictive coding, where neurons attempt to locally predict their neighbors in a specific pattern corresponding to their neighbors known structured selectivies.

We note that while the results presented in Chapter 9 are promising and show that indeed Homomorphic SSL is equivalent to standard augmentation-based SSL for the settings we tested, we did not yet test the ability for such a learning algorithm to be

combined with localized learning similar to Greedy Infomax. Therefore, while our results provide a blueprint for how a modern local self-supervised learning algorithm may be constructed, and additionally provide evidence that is may be successful, a definitive answer will still require further experimentation and empirical support.

10.4. Future Work: The Unanswered Questions

As with any scientific endeavor, a true measure of success should not only be based on which questions are answered, but also on which new questions are brought up in the process. In completing this work, the following questions have come up, and remained on our minds unanswered. We therefore propose them as what we believe to be valuable directions for future work.

Research Question 5: Can topographic generative models with multiple levels of latent variables simultaneously explain the topographic organization of multiple levels of the hierarchical visual processing stream?

In Chapter 4, we demonstrated how topographic organization could be induced in a single level of latent variables at the end of a deep neural network feature extractor. In future work, it would be interesting to explore if hierarchical latent variable models such as the NVAE (Vahdat and Kautz, 2020) of VDVAE (Child, 2021) could be adapted to incorporate our topographic prior and thus represent topographic organization at multiple levels of abstraction. If such models were to be able to successfully model a greater range of topographic organization, this would be promising evidence for the 'Bayesian brain' hypothesis (Helmholtz, 1948) and the interpretation of natural neural networks as instantiations of probabilistic generative models.

Research Question 6: To what extent can traveling wave dynamics explain topographic organization in the brain?

As we saw in Chapter 6, it is clear that in our model, traveling waves did appear to induce a form of topographic organization reminiscent of that found in the early visual system of many mammals. However, it is not clear how much further this mechanism will extend. In future work it would be interesting to explore if there is a more fundamental connection between topographic organization and traveling wave activity that may help to describe the ubiquity of both observations in the brain.

Research Question 7: What are the new mathematical descriptions of structured representations, beyond equivariance, that we may be able to derive from observa-

tions of natural structure?

In this work we have demonstrated multiple natural mechanisms which induce a form of approximate equivariance. However, given the non-group structure of the transformations which we intend for these models to capture, it is no longer consistent to call such models ‘equivariant’. In future work, we therefore aim to identify more precisely which alternative mathematical structures may be better descriptions of the types of structured representations we see in natural neural activity. In Chapter 8 we have put forth one potential form of such a mathematical structure through the framework of optimal transport and gradient flows, however we believe there is still significant work to be done on this front and are excited to continue down this path in future work. Ultimately, these ideas are reminiscent of the quote from John von Neumann at the forefront of this thesis: "I suspect that a deeper mathematical study of the nervous system ... may alter the way in which we look at mathematics and logics proper."

Research Question 8: What is the ideal decomposition of the world which naturally facilitates generalization and robustness?

As we said before, we believe the brain has found such a decomposition through millions of years of evolution, and we believe that it is likely this decomposition that is what allows for natural human generalization with limited data. In future work, we believe one of the ultimate goals of our research agenda should be to discover this decomposition and any necessary biological mechanisms which allow for its perpetuation.

In conclusion, we thank the audience for reading, and hope that this work has, at least in a small way, encouraged future work to understand the true beauty behind the nature of intelligence.

References

BIBLIOGRAPHY

Ackman, James B., Timothy J. Burbridge, and Michael C. Crair (Oct. 2012). "Retinal waves coordinate patterned activity throughout the developing visual system". In: *Nature* 490.7419, pp. 219–225. DOI: 10.1038/nature11529. URL: https://doi.org/10.1038/nature11529.

Afifi, Mahmoud (2019). "11K Hands: gender recognition and biometric identification using a large dataset of hand images". In: *Multimedia Tools and Applications*. DOI: 10.1007/s11042-019-7424-8. URL: https://doi.org/10.1007/s11042-019-7424-8.

Agrawal, Pulkit, Joao Carreira, and Jitendra Malik (2015). "Learning to see by moving". In: *ICCV*.

Alamia, Andrea and Rufin VanRullen (Oct. 2019). "Alpha oscillations and traveling waves: Signatures of predictive coding?" In: *PLOS Biology* 17.10. Ed. by Adam Kohn, e3000487. DOI: 10.1371/journal.pbio.3000487. URL: https://doi.org/10.1371/journal.pbio.3000487.

Alink, Arjen, Caspar M. Schwiedrzik, Axel Kohler, Wolf Singer, and Lars Muckli (2010). "Stimulus Predictability Reduces Responses in Primary Visual Cortex". In: *Journal of Neuroscience* 30.8, pp. 2960–2966. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.3730-10.2010. eprint: https://www.jneurosci.org/content/30/8/2960.full.pdf. URL: https://www.jneurosci.org/content/30/8/2960.

Aparicio, Paul L., Elias B. Issa, and James J. DiCarlo (2016). "Neurophysiological Organization of the Middle Face Patch in Macaque Inferior Temporal Cortex". In: *Journal of Neuroscience* 36.50, pp. 12729–12745. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.0237-16.2016. eprint: https://www.jneurosci.org/content/36/50/12729.full.pdf. URL: https://www.jneurosci.org/content/36/50/12729.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein generative adversarial networks". In: *ICML*.

Arjovsky, Martin, Amar Shah, and Yoshua Bengio (2016). "Unitary Evolution Recurrent Neural Networks". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 1120–1128.

Arora, Sanjeev, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi (2019). *A Theoretical Analysis of Contrastive Unsupervised Representation Learning*. DOI: 10.48550/ARXIV.1902.09229. URL: https://arxiv.org/abs/1902.09229.

Ba, Jimmy, Geoffrey Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu (2016). *Using Fast Weights to Attend to the Recent Past*. arXiv: 1610.06258 [stat.ML].

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. DOI: 10.48550/ARXIV.1607.06450. URL: https://arxiv.org/abs/1607.06450.

Bachman, Philip, R Devon Hjelm, and William Buchwalter (2019). "Learning Representations by Maximizing Mutual Information Across Views". In: *arXiv preprint arXiv:1906.00910*.

Bajaj, Chandrajit, Luke McLennan, Timothy Andeen, and Avik Roy (2023). "Recipes for when Physics Fails: Recovering Robust Learning of Physics Informed Neural Networks". In: *Machine Learning: Science and Technology*.

Bakhtiari, Shahab, Patrick Mineault, Tim Lillicrap, Christopher C. Pack, and Blake A. Richards (2021). "The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning". In: *bioRxiv*. DOI: 10.1101/2021.06.18.448989. eprint: https://www.biorxiv.org/content/early/2021/06/24/2021.06.18.448989.full.pdf. URL: https://www.biorxiv.org/content/early/2021/06/24/2021.06.18.448989.

Barlow, Horace B et al. (1961). "Possible principles underlying the transformation of sensory messages". In: *Sensory communication* 1.01.

Bell, Anthony J. and Terrence J. Sejnowski (Nov. 1995). "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". In: *Neural Computation* 7.6, pp. 1129–1159. ISSN: 0899-7667. DOI: 10.1162/neco.1995.7.6.1129. eprint: https://direct.mit.edu/neco/article-pdf/7/6/1129/813064/neco.1995.7.6.1129.pdf. URL: https://doi.org/10.1162/neco.1995.7.6.1129.

Benamou, Jean-David and Yann Brenier (2000). "A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem". In: *Numerische Mathematik*.

Bengio, Yoshua (2014). "How Auto-Encoders Could Provide Credit Assignment in Deep Networks via Target Propagation". In: *CoRR* abs/1407.7906. arXiv: 1407.7906.

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

Bengio, Yoshua, Nicholas Léonard, and Aaron Courville (2013). "Estimating or propagating gradients through stochastic neurons for conditional computation". In: *arXiv preprint arXiv:1308.3432*.

Benigno, Gabriel, Roberto Budzinski, Zachary Davis, John Reynolds, and Lyle Muller (Aug. 2022). "Waves traveling over a map of visual space can ignite short-term predictions of sensory input". In: *Research Square Platform LLC*. DOI: 10.21203/rs.3.rs-1903144/v1. URL: https://doi.org/10.21203/rs.3.rs-1903144/v1.

Benton, Gregory, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson (2020). "Learning invariances in neural networks". English (US). In: *Advances in Neural Information Processing Systems* December. ISSN: 1049-5258.

Berger, Hans (Dec. 1929). "Über das Elektrenkephalogramm des Menschen". In: *Archiv für Psychiatrie und Nervenkrankheiten* 87.1, pp. 527–570. DOI: 10.1007/bf01797193. URL: https://doi.org/10.1007/bf01797193.

Besserve, Michel, Nikos Logothetis, and Bernhard Schölkopf (Jan. 2015). "Shifts of Gamma Phase across Primary Visual Cortical Sites Reflect Dynamic Stimulus-Modulated Information Transfer". In: DOI: 10.15496/publikation-10582.

Bhattacharya, Sayak, Scott L. Brincat, Mikael Lundqvist, and Earl K. Miller (Jan. 2022). "Traveling waves in the prefrontal cortex during working memory". In: *PLOS Computational Biology* 18.1, pp. 1–22. DOI: 10.1371/journal.pcbi.1009827. URL: https://doi.org/10.1371/journal.pcbi.1009827.

Bhattacharya, Sayak, Jacob A Donoghue, Meredith Mahnke, Scott L Brincat, Emery N Brown, and Earl K Miller (2022). "Propofol anesthesia alters cortical traveling waves". In: *Journal of Cognitive Neuroscience* 34.7, pp. 1274–1286.

Bishop, Christopher, Markus Svensen, and Christopher Williams (May 1997). "GTM: The Generative Topographic Mapping". In: *Neural Computation* 10, pp. 215–234. DOI: 10.1162/089976698300017953.

Blaas, Arno, Xavier Suau, Jason Ramapuram, Nicholas Apostoloff, and Luca Zappella (13 Dec 2021). "Challenges of Adversarial Image Augmentations". In: *Proceedings on "I (Still) Can't Believe It's Not Better!" at NeurIPS 2021 Workshops*. Ed. by Melanie F. Pradier, Aaron Schein, Stephanie Hyland, Francisco J. R. Ruiz, and Jessica Z. Forde. Vol. 163. Proceedings of Machine Learning Research. PMLR, pp. 9–14. URL: https://proceedings.mlr.press/v163/blaas22a.html.

Blauch, Nicholas M., Marlene Behrmann, and David C. Plaut (2021). "A connectivity-constrained computational account of topographic organization in primate high-level visual cortex". In: *bioRxiv*. DOI: 10.1101/2021.05.29.446297. eprint: https://www.biorxiv.org/content/early/2021/07/12/2021.05.29.446297.full.pdf. URL: https://www.biorxiv.org/content/early/2021/07/12/2021.05.29.446297.

Boenninghoff, Benedikt, Steffen Zeiler, Robert M. Nickel, and Dorothea Kolossa (2020). "Variational Autoencoder with Embedded Student-*t* Mixture Model for Authorship Attribution". In: *ArXiv* abs/2005.13930.

Bordelon, Blake and Cengiz Pehlevan (Dec. 2022). "Population codes enable learning from few examples by shaping inductive bias". In: *eLife* 11. Ed. by Thomas Serre, Michael J Frank, Fabio Anselmi, and Jeff Beck, e78606. ISSN: 2050-084X. DOI: 10.7554/eLife.78606. URL: https://doi.org/10.7554/eLife.78606.

Botev, Aleksandar, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins (2021). *Which priors matter? Benchmarking models for learning latent dynamics*. DOI: 10.48550/ARXIV.2111.05458. URL: https://arxiv.org/abs/2111.05458.

Bouchacourt, Diane, Mark Ibrahim, and Stéphane Deny (2021). "Addressing the Topological Defects of Disentanglement via Distributed Operators". In: *ArXiv* abs/2102.05623.

Bragin, Anatol, Gábor Jandó, Zoltán Nádasdy, Jamille Hetke, Kensall Wise, and Gy Buzsáki (1995). "Gamma (40-100 Hz) oscillation in the hippocampus of the behaving rat". In: *Journal of neuroscience* 15.1, pp. 47–60.

Brandstetter, Johannes, Daniel Worrall, and Max Welling (2022). "Message passing neural PDE solvers". In: *ICLR*.

Breakspear, Michael, Stewart Heitmann, and Andreas Daffertshofer (2010). "Generative Models of Cortical Oscillations: Neurobiological Implications of the Kuramoto Model". In: *Frontiers in Human Neuroscience* 4. ISSN: 1662-5161. DOI: 10.3389/fnhum.2010.00190. URL: https://www.frontiersin.org/articles/10.3389/fnhum.2010.00190.

Brysbaert, Marc (2019). "How many words do we read per minute? A review and meta-analysis of reading rate". In: *Journal of Memory and Language* 109, p. 104047. ISSN: 0749-596X. DOI: https://doi.org/10.1016/j.jml.2019.104047. URL: https://www.sciencedirect.com/science/article/pii/S0749596X19300786.

Burgess, Chris and Hyunjik Kim (2018). *3D Shapes Dataset*. URL: https://github.com/deepmind/3dshapes-dataset/.

Cadieu, Charles F., Ha Hong, Daniel L. K. Yamins, Nicolas Pinto, Diego Ardila, Ethan A. Solomon, Najib J. Majaj, and James J. DiCarlo (Dec. 2014). "Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition". In: *PLOS Computational Biology* 10.12, pp. 1–18. DOI: 10.1371/journal.pcbi.1003963. URL: https://doi.org/10.1371/journal.pcbi.1003963.

Cagigal, M. P., L. Vega, and P. Prieto (Apr. 1995). "Movement characterization with the spatiotemporal Fourier transform of low-light-level images". In: *Applied Optics* 34.11, p. 1769. DOI: 10.1364/ao.34.001769. URL: https://doi.org/10.1364/ao.34.001769.

Cao, Qiong, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman (2018). "Vggface2: A dataset for recognising faces across pose and age". In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, pp. 67–74.

Caspers, Julian, Nicola Palomero-Gallagher, Svenja Caspers, Axel Schleicher, Katrin Amunts, and Karl Zilles (Oct. 2013). "Receptor architecture of visual areas in the face and word-form recognition region of the posterior fusiform gyrus". In: *Brain structure & function* 220. DOI: 10.1007/s00429-013-0646-z.

Caton, Richard (1875). "The electric currents of the brain". In: *British Medical Journal* 2, p. 278.

Caton, Richard (1877). "Interim report on investigations of the electric currents of the brain." In: *British Medical Journal* 1, pp. 62–65.

Cesa, Gabriele, Leon Lang, and Maurice Weiler (2022). "A Program to Build E(N)-Equivariant Steerable CNNs". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=WE4qe9xlnQw.

Chang, Bo, Minmin Chen, Eldad Haber, and Ed H. Chi (2019). "AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=ryxepo0cFX.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud (2018). *Neural Ordinary Differential Equations*. DOI: 10.48550/ARXIV.1806.07366. URL: https://arxiv.org/abs/1806.07366.

Chen, Ricky TQ, Xuechen Li, Roger B Grosse, and David K Duvenaud (2018). "Isolating sources of disentanglement in variational autoencoders". In: *NeurIPS*.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020). "A Simple Framework for Contrastive Learning of Visual Representations". In: *arXiv preprint arXiv:2002.05709*.

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel (2016). "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *NeurIPS*.

Chen, Xinlei and Kaiming He (2020). *Exploring Simple Siamese Representation Learning*. arXiv: 2011.10566 [cs.CV].

Chen, Yusi, Huanqiu Zhang, and Terrence J. Sejnowski (2022). "Hippocampus as a generative circuit for predictive coding of future sequences". In: *bioRxiv*. DOI: 10.1101/2022.05.19.492731. eprint: https://www.biorxiv.org/content/early/2022/05/20/2022.05.19.492731.full.pdf. URL: https://www.biorxiv.org/content/early/2022/05/20/2022.05.19.492731.

Chen, Z. and H. Zhang (June 2019). "Learning Implicit Fields for Generative Shape Modeling". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 5932–5941. DOI: 10.1109/CVPR.2019.00609. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00609.

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

Chilkuri, Narsimha Reddy and Chris Eliasmith (18–24 Jul 2021). "Parallelizing Legendre Memory Unit Training". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 1898–1907. URL: https://proceedings.mlr.press/v139/chilkuri21a.html.

Chizat, Lenaic, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré (2020). "Faster wasserstein distance estimation with the sinkhorn divergence". In: *NeurIPS*.

Cichy, Radoslaw Martin, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva (2016). "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence". In: *Scientific Reports* 6.1, p. 27755. DOI: 10.1038/srep27755. URL: https://doi.org/10.1038/srep27755.

Clark, Andy (2013). "Whatever next? Predictive brains, situated agents, and the future of cognitive science". In: *Behavioral and Brain Sciences* 36.3, pp. 181–204. DOI: 10.1017/S0140525X12000477.

Clemons, william (Dec. 2018). "Human body identification and verification dataset". In: *Mendeley Data*.

Cohen, Jack (1988). *Statistical Power Analysis for the behavioral sciences*. L. Erlbaum Associates.

Cohen, Taco and M. Welling (2017). "Steerable CNNs". In: *ArXiv* abs/1612.08498.

Cohen, Taco and Max Welling (22–24 Jun 2014). "Learning the Irreducible Representations of Commutative Lie Groups". In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1755–1763. URL: http://proceedings.mlr.press/v32/cohen14.html.

Cohen, Taco and Max Welling (2016a). "Group equivariant convolutional networks". In: *International conference on machine learning*, pp. 2990–2999.

Cohen, Taco and Max Welling (2016b). "Group equivariant convolutional networks". In: *International conference on machine learning*, pp. 2990–2999.

Cohen, Taco S and Max Welling (2017). "Steerable cnns". In: *ICLR*.

Cohen, Taco S. and Max Welling (2015). "Transformation Properties of Learned Visual Representations". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1412.7659.

Comon, Pierre (1994). "Independent component analysis, a new concept?" In: *Signal processing* 36.3, pp. 287–314.

Connor, Marissa, Gregory Canal, and Christopher Rozell (13–15 Apr 2021). "Variational Autoencoder with Learned Latent Structure". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 2359–2367. URL: http://proceedings.mlr.press/v130/connor21a.html.

Conwell, Colin, Jacob S. Prince, Kendrick N. Kay, George A. Alvarez, and Talia Konkle (2023). "What can 1.8 billion regressions tell us about the pressures shaping high-level visual representation in brains and machines?" In: *bioRxiv*. DOI: 10.1101/2022.03.28.485868. eprint: https://www.biorxiv.org/content/early/2023/07/01/2022.03.28.485868.full.pdf. URL: https://www.biorxiv.org/content/early/2023/07/01/2022.03.28.485868.

Cuturi, Marco (2013). "Sinkhorn distances: Lightspeed computation of optimal transport". In: *NeurIPS*.

Dangovski, Rumen, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic (2022). "Equivariant Self-Supervised Learning: Encouraging Equivariance in Representations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=gKLAAfiytI.

Dapello, Joel, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D. Cox, and James J. DiCarlo (2020). "Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations". In: *bioRxiv*. DOI: 10.1101/2020.06.16.154542. eprint: https://www.biorxiv.org/content/early/2020/06/17/2020.06.16.154542.full.pdf. URL: https://www.biorxiv.org/content/early/2020/06/17/2020.06.16.154542.

Davis, Zachary W., Gabriel B. Benigno, Charlee Fletterman, Theo Desbordes, Christopher Steward, Terrence J. Sejnowski, John H. Reynolds, and Lyle Muller (Oct. 2021). "Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states". In: *Nature Communications* 12.1. DOI: 10.1038/s41467-021-26175-1. URL: https://doi.org/10.1038/s41467-021-26175-1.

Davis, Zachary W., Lyle Muller, Julio Martinez-Trujillo, Terrence Sejnowski, and John H. Reynolds (Oct. 2020). "Spontaneous travelling cortical waves gate perception in behaving primates". In: *Nature* 587.7834, pp. 432–436. DOI: 10.1038/

s41586-020-2802-y. URL: https://doi.org/10.1038/s41586-020-2802-y.

Dayan, Peter, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel (Sept. 1995). "The Helmholtz Machine". In: *Neural Computation* 7.5, pp. 889–904. DOI: 10.1162/neco.1995.7.5.889. URL: https://doi.org/10.1162/neco.1995.7.5.889.

Dehmamy, Nima, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu (2021). "Automatic Symmetry Discovery with Lie Algebra Convolutional Network". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. URL: https://openreview.net/forum?id=NPOWF_ZLfC5.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

DeVries, Terrance and Graham W. Taylor (2017). *Dataset Augmentation in Feature Space*. DOI: 10.48550/ARXIV.1702.05538. URL: https://arxiv.org/abs/1702.05538.

Dey, Neel, Antong Chen, and Soheil Ghafurian (2021). "Group equivariant generative adversarial networks". In: *ICLR*.

Diaconu, Nichita and Daniel Worrall (Sept. 2019a). "Learning to Convolve: A Generalized Weight-Tying Approach". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 1586–1595. URL: https://proceedings.mlr.press/v97/diaconu19a.html.

Diaconu, Nichita and Daniel Worrall (2019b). "Learning to convolve: A generalized weight-tying approach". In: *ICML*. PMLR.

Diamant, NE and A Bortoff (Feb. 1969). "Nature of the intestinal low-wave frequency gradient". In: *American Journal of Physiology-Legacy Content* 216.2, pp. 301–307. DOI: 10.1152/ajplegacy.1969.216.2.301. URL: https://doi.org/10.1152/ajplegacy.1969.216.2.301.

Dilokthanakul, Nat, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan (2016). "Deep unsupervised clustering with gaussian mixture variational autoencoders". In: *ICLR*.

Ding, Zheng, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu (2020). "Guided variational autoencoder for disentanglement learning". In: *CVPR*.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2017). "Density estimation using real nvp". In: *ICLR*.

Dobs, Katharina, Julio Martinez, Alexander J.E. Kell, and Nancy Kanwisher (2021). "Brain-like functional specialization emerges spontaneously in deep neural networks". In: *bioRxiv*. DOI: 10.1101/2021.07.05.451192. eprint: https://www.biorxiv.org/content/early/2021/07/06/2021.07.05.451192.full.pdf. URL: https://www.biorxiv.org/content/early/2021/07/06/2021.07.05.451192.

Doshi, Fenil R. and Talia Konkle (2022). "Visual object topographic motifs emerge from self-organization of a unified representational space". In: *bioRxiv*. DOI: 10.1101/2022.09.06.506403. eprint: https://www.biorxiv.org/content/early/2022/09/08/2022.09.06.506403.full.pdf. URL: https://www.biorxiv.org/content/early/2022/09/08/2022.09.06.506403.

Dupont, Emilien (2018). "Learning disentangled joint continuous and discrete representations". In: *NeurIPS*.

Eastwood, Cian and Christopher KI Williams (2018). "A framework for the quantitative evaluation of disentangled representations". In: *ICLR*.

Egner, Tobias, Jim Monti, and Christopher Summerfield (Dec. 2010). "Expectation and Surprise Determine Neural Population Responses in the Ventral Visual Stream". In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 30, pp. 16601–8. DOI: 10.1523/JNEUROSCI.2770-10.2010.

Eisenberger, Marvin, Aysim Toker, Laura Leal-Taixé, Florian Bernard, and Daniel Cremers (2022). "A Unified Framework for Implicit Sinkhorn Differentiation". In: *CVPR*.

Elesedy, Bryn and Sheheryar Zaidi (18–24 Jul 2021). "Provably Strict Generalisation Benefit for Equivariant Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 2959–2969. URL: https://proceedings.mlr.press/v139/elesedy21a.html.

Elias, P. (1955). "Predictive coding-I". In: *IRE Trans. Inf. Theory* 1, pp. 16–24.

Elman, Jeffrey L. (1990). "Finding Structure in Time". In: *Cognitive Science* 14.2, pp. 179–211. DOI: https://doi.org/10.1207/s15516709cog1402_1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

Epstein, Russell A. and Nancy Kanwisher (1998). "A cortical representation of the local visual environment". In: *Nature* 392, pp. 598–601.

Erichson, N. Benjamin, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney (2021). "Lipschitz Recurrent Neural Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=-N7PBXqOUJZ.

Ermentrout, Bard, Alla Borisyuk, Avner Friedman, and David Terman (Jan. 1970). "Neural Oscillators". In: vol. 1860, pp. 69–106. ISBN: 978-3-540-23858-4. DOI: 10.1007/978-3-540-31544-5_3.

Ermentrout, G Bard and David Kleinfeld (2001). "Traveling electrical waves in cortex: insights from phase dynamics and speculation on a computational role". In: *Neuron* 29.1, pp. 33–44.

Ermentrout, George Bard and Nancy Kopell (1984). "Frequency plateaus in a chain of weakly coupled oscillators, I." In: *SIAM journal on Mathematical Analysis* 15.2, pp. 215–237.

Essen, David C. Van (1997). "A tension-based theory of morphogenesis and compact wiring in the central nervous system". In: *Nature* 385.6614, pp. 313–318. DOI: 10.1038/385313a0. URL: https://doi.org/10.1038/385313a0.

Farrell, Matthew, Blake Bordelon, Shubhendu Trivedi, and Cengiz Pehlevan (2021). *Capacity of Group-invariant Linear Readouts from Equivariant Representations: How Many Objects can be Linearly Classified Under All Possible Views?* DOI: 10.48550/ARXIV.2110.07472. URL: https://arxiv.org/abs/2110.07472.

Federici, Marco, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata (2020). *Learning Robust Representations via Multi-View Information Bottleneck*. DOI: 10.48550/ARXIV.2002.07017. URL: https://arxiv.org/abs/2002.07017.

Ferrier, David (Dec. 1874). "The localization of function in the brain". In: *Proceedings of the Royal Society of London* 22.148-155, pp. 228–232. DOI: 10.1098/rspl.1873.0032. URL: https://doi.org/10.1098/rspl.1873.0032.

Feydy, Jean, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré (2019). "Interpolating between optimal transport and mmd using sinkhorn divergences". In: *AISTATS*.

Finlay, Chris, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman (2020). "How to train your neural ODE: the world of Jacobian and kinetic regularization". In: *ICML*. PMLR.

Finzi, Marc, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson (13–18 Jul 2020). "Generalizing Convolutional Neural Networks for Equivariance to Lie Groups on Arbitrary Continuous Data". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 3165–3176. URL: http://proceedings.mlr.press/v119/finzi20a.html.

Finzi, Marc, Max Welling, and Andrew Gordon Gordon Wilson (18–24 Jul 2021). "A Practical Method for Constructing Equivariant Multilayer Perceptrons for Arbitrary Matrix Groups". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 3318–3328. URL: https://proceedings.mlr.press/v139/finzi21a.html.

Földiák, Peter (June 1991). "Learning Invariance From Transformation Sequences". In: *Neural Computation* 3, pp. 194–200. DOI: 10.1162/neco.1991.3.2.194.

Friston, Karl (May 2005). "A Theory of Cortical Responses". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 360, pp. 815–36. DOI: 10.1098/rstb.2005.1622.

Friston, Karl J. (Oct. 2019). "Waves of prediction". In: *PLOS Biology* 17.10, e3000426. DOI: 10.1371/journal.pbio.3000426. URL: https://doi.org/10.1371/journal.pbio.3000426.

Frogner, Charlie, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio (2015). "Learning with a Wasserstein loss". In: *NeurIPS*.

Fukushima, Kunihiko (Apr. 1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4, pp. 193–202. DOI: 10.1007/bf00344251. URL: https://doi.org/10.1007/bf00344251.

Gefter, Amanda (Jan. 2015). *The man who tried to redeem the world with logic*. URL: https://nautil.us/the-man-who-tried-to-redeem-the-world-with-logic-235253/.

Goetschalckx, Lore, Alex Andonian, Aude Oliva, and Phillip Isola (2019). "Ganalyze: Toward visual definitions of cognitive image properties". In: *ICCV*.

Gong, Pulin and Cees van Leeuwen (Dec. 2009). "Distributed Dynamical Computation in Neural Circuits with Propagating Coherent Activity Patterns". In: *PLOS Computational Biology* 5.12, pp. 1–11. DOI: 10.1371/journal.pcbi.1000611. URL: https://doi.org/10.1371/journal.pcbi.1000611.

Grathwohl, Will and Aaron Wilson (2016). "Disentangling Space and Time in Video with Hierarchical Variational Auto-encoders". In: *CoRR* abs/1612.04440. arXiv: 1612.04440. URL: http://arxiv.org/abs/1612.04440.

Graves, Alex, Greg Wayne, and Ivo Danihelka (2014). *Neural Turing Machines*. arXiv: 1410.5401 [cs.NE].

Graves, Alex, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis (Oct. 2016). "Hybrid computing using a neural network with dynamic external memory". In: *Nature* 538.7626, pp. 471–476. DOI: 10.1038/nature20101. URL: https://doi.org/10.1038/nature20101.

Grill, Jean-Bastien, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and

Michal Valko (2020). *Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning*. arXiv: 2006.07733 [cs.LG].

Grill-Spector, Kalanit and Kevin S. Weiner (2014). "The functional architecture of the ventral temporal cortex and its role in categorization". In: *Nature Reviews Neuroscience* 15.8, pp. 536–548. DOI: 10.1038/nrn3747. URL: https://doi.org/10.1038/nrn3747.

Gu, Albert, Karan Goel, and Christopher Re (2022). "Efficiently Modeling Long Sequences with Structured State Spaces". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=uYLFoz1vlAC.

Gu, Albert, Caglar Gulcehre, Tom Le Paine, Matt Hoffman, and Razvan Pascanu (2020). *Improving the Gating Mechanism of Recurrent Neural Networks*. arXiv: 1910.09890 [cs.NE].

Güçlü, Umut and Marcel A. J. van Gerven (2015). "Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream". In: *Journal of Neuroscience* 35.27, pp. 10005–10014. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5023-14.2015. eprint: https://www.jneurosci.org/content/35/27/10005.full.pdf. URL: https://www.jneurosci.org/content/35/27/10005.

Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris (2020). "Ganspace: Discovering interpretable gan controls". In: *NeurIPS*.

Haxby, James, Maria Gobbini, Maura Furey, Alumit Ishai, Jennifer Schouten, and Pietro Pietrini (Oct. 2001). "Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex". In: *Science (New York, N.Y.)* 293, pp. 2425–30. DOI: 10.1126/science.1063736.

Haxby, James, Jyothi Swaroop Guntupalli, Andrew Connolly, Yaroslav Halchenko, Bryan Conroy, Maria Gobbini, Michael Hanke, and Peter Ramadge (Oct. 2011). "A Common, High-Dimensional Model of the Representational Space in Human Ventral Temporal Cortex". In: *Neuron* 72, pp. 404–16. DOI: 10.1016/j.neuron.2011.08.026.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. arXiv: 1502.01852 [cs.CV].

Hebb, Donald (1949). *The organization of behavior: A neuropsychological theory*.

Heitmann, Stewart and G. Bard Ermentrout (Sept. 2020). "Direction-selective motion discrimination by traveling waves in visual cortex". In: *PLOS Computational Biology* 16.9, pp. 1–20. DOI: 10.1371/journal.pcbi.1008164. URL: https://doi.org/10.1371/journal.pcbi.1008164.

Helmholtz, Hermann von (1948). "Concerning the perceptions in general, 1867." In: *Readings in the history of psychology*. Appleton-Century-Crofts, pp. 214–230. DOI: 10.1037/11304-027. URL: https://doi.org/10.1037/11304-027.

Henaff, Mikael, Arthur Szlam, and Yann LeCun (20–22 Jun 2016). "Recurrent Orthogonal Networks and Long-Memory Tasks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 2034–2042. URL: https://proceedings.mlr.press/v48/henaff16.html.

Hendrycks, Dan, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer (2020). "The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization". In: *CoRR* abs/2006.16241. arXiv: 2006.16241. URL: https://arxiv.org/abs/2006.16241.

Higgins, Irina, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner (2018). "Towards a Definition of Disentangled Representations". In: *ArXiv* abs/1812.02230.

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner (2016). "beta-vae: Learning basic visual concepts with a constrained variational framework". In: *ICLR*.

Higgins, Irina, Peter Wirnsberger, Andrew Jaegle, and Aleksandar Botev (2021). "SyMetric: Measuring the Quality of Learnt Hamiltonian Dynamics Inferred from Vision". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 25591–25605. URL: https://proceedings.neurips.cc/paper/2021/file/d6ef5f7fa914c19931a55bb262ec879c-Paper.pdf.

Hinton, Geoffrey E, Alex Krizhevsky, and Sida D Wang (2011a). "Transforming auto-encoders". In: *ICANN*. Springer.

Hinton, Geoffrey E, Sara Sabour, and Nicholas Frosst (2018). "Matrix capsules with EM routing". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HJWLfGWRb.

Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang (2011b). "Transforming Auto-Encoders". In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 44–51.

Hinton, Geoffrey E. and Yee-Whye Teh (2001). "Discovering Multiple Constraints That Are Frequently Approximately Satisfied". In: *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. UAI'01. Seattle, Washington, pp. 227–234. ISBN: 1558608001.

Hjelm, R Devon, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio (2019). "Learning deep representations by mutual information estimation and maximization". In: *International*

Conference on Learning Representations. URL: https://openreview.net/forum?id=Bklr3j0cKX.

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

Hoogeboom, Emiel, Vıctor Garcia Satorras, Clément Vignac, and Max Welling (2022). "Equivariant diffusion for molecule generation in 3d". In: *ICML*. PMLR.

Hsieh, Jun-Ting, Shengjia Zhao, Stephan Eismann, Lucia Mirabella, and Stefano Ermon (2019). "Learning neural PDE solvers with convergence guarantees". In: *ICLR*.

Huang, Gary B., Manu Ramesh, Tamara Berg, and Erik Learned-Miller (Oct. 2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst.

Hubel, D. H. and T. N. Wiesel (Oct. 1959). "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of Physiology* 148.3, pp. 574–591. DOI: 10.1113/jphysiol.1959.sp006308. URL: https://doi.org/10.1113/jphysiol.1959.sp006308.

Hubel, D. H. and T. N. Wiesel (Jan. 1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160.1, pp. 106–154. DOI: 10.1113/jphysiol.1962.sp006837. URL: https://doi.org/10.1113/jphysiol.1962.sp006837.

Hubel, David H and Torsten N Wiesel (1974a). "Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor". In: *Journal of Comparative Neurology* 158.3, pp. 295–305.

Hubel, David H, Torsten N Wiesel, and Michael P Stryker (1978). "Anatomical demonstration of orientation columns in macaque monkey". In: *Journal of Comparative Neurology* 177.3, pp. 361–379.

Hubel, David H. and Torsten N. Wiesel (1974b). "Sequence regularity and geometry of orientation columns in the monkey striate cortex". In: *Journal of Comparative Neurology* 158.3, pp. 267–293. DOI: https://doi.org/10.1002/cne.901580304. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.901580304. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.901580304.

Hughes, Dr. John R. (1995). "The Phenomenon of Travelling Waves: A Review". In: *Clinical Electroencephalography* 26.1, pp. 1–6. DOI: 10.1177/155005949502600103. URL: https://doi.org/10.1177/155005949502600103.

Hurri, Jarmo and Aapo Hyvärinen (Mar. 2003). "Simple-Cell-Like Receptive Fields Maximize Temporal Coherence in Natural Video". In: *Neural Computation* 15.3, pp. 663–691. ISSN: 0899-7667. DOI: 10.1162/089976603321192121. URL: https://doi.org/10.1162/089976603321192121.

Huth, Alexander G., Wendy A. de Heer, Thomas L. Griffiths, Frédéric E. Theunissen, and Jack L. Gallant (2016). "Natural speech reveals the semantic maps that tile human cerebral cortex". In: *Nature* 532.7600, pp. 453–458. DOI: 10.1038/nature17637. URL: https://doi.org/10.1038/nature17637.

Hyvärinen, A., J. Hurri, and Jaakko J. Väyrynen (2004). "A unifying framework for natural image statistics: spatiotemporal activity bubbles". In: *Neurocomputing* 58-60, pp. 801–806.

Hyvärinen, Aapo (1998). "Independent component analysis in the presence of gaussian noise by maximizing joint likelihood". In: *Neurocomputing* 22.1-3, pp. 49–67.

Hyvärinen, Aapo and Patrik Hoyer (2000). "Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces". In: *Neural computation* 12.7, pp. 1705–1720.

Hyvärinen, Aapo, Patrik O Hoyer, and Mika Inki (2001). "Topographic independent component analysis". In: *Neural computation* 13.7, pp. 1527–1558.

Hyvärinen, Aapo and Patrik O. Hoyer (2001). "A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images". In: *Vision Research* 41.18, pp. 2413–2423. ISSN: 0042-6989. DOI: https://doi.org/10.1016/S0042-6989(01)00114-6. URL: https://www.sciencedirect.com/science/article/pii/S0042698901001146.

Hyvärinen, Aapo, Jarmo Hurri, and Patrick O Hoyer (2009). *Natural image statistics: A probabilistic approach to early computational vision.* Vol. 39. Springer Science & Business Media.

Hyvärinen, Aapo and Erkki Oja (2000). "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5, pp. 411–430.

Izhikevich, Eugene M and Frank C. Hoppensteadt (Jan. 2008). "Polychronous Wavefront Computations". In: *International Journal of Bifurcation and Chaos* 19.5, pp. 1733–1739.

Jaderberg, Max, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu (June 2017). "Decoupled Neural Interfaces using Synthetic Gradients". In: vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 1627–1635.

Jahanian, Ali, Lucy Chai, and Phillip Isola (2020). "On the" steerability" of generative adversarial networks". In: *ICLR*.

Jancke, Dirk, Frédéric Chavane, Shmuel Naaman, and Amiram Grinvald (2004). "Imaging cortical correlates of illusion in early visual cortex". In: *Nature* 428.6981, pp. 423–426.

Jang, Eric, Shixiang Gu, and Ben Poole (2017). "Categorical reparameterization with gumbel-softmax". In: *ICLR*.

Jeong, Seong-Ok, Tae-Wook Ko, and Hie-Tae Moon (Sept. 2002). "Time-Delayed Spatial Patterns in a Two-Dimensional Array of Coupled Oscillators". In: *Phys. Rev. Lett.* 89 (15), p. 154104. DOI: 10.1103/PhysRevLett.89.154104. URL: https://link.aps.org/doi/10.1103/PhysRevLett.89.154104.

Jeong, Yeonwoo and Hyun Oh Song (2019). "Learning discrete and continuous factors of data via alternating disentanglement". In: *ICML*.

Ji, Wenlong, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang (2021). *The Power of Contrast for Feature Learning: A Theoretical Analysis*. arXiv: 2110.02473 [cs.LG].

Journé, Adrien, Hector Garcia Rodriguez, Qinghai Guo, and Timoleon Moraitis (2022). *Hebbian Deep Learning Without Feedback*. arXiv: 2209.11883 [cs.NE].

Kag, Anil, Ziming Zhang, and Venkatesh Saligrama (2020). "RNNs Incrementally Evolving on an Equilibrium Manifold: A Panacea for Vanishing and Exploding Gradients?" In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HylpqA4FwS.

Kaiser, Łukasz and Ilya Sutskever (2016). *Neural GPUs Learn Algorithms*. arXiv: 1511.08228 [cs.LG].

Kanwisher, Nancy, Meenakshi Khosla, and Katharina Dobs (2023). "Using artificial neural networks to ask 'why' questions of minds and brains". In: *Trends in Neurosciences*. ISSN: 0166-2236. DOI: https://doi.org/10.1016/j.tins.2022.12.008. URL: https://www.sciencedirect.com/science/article/pii/S0166223622002624.

Kanwisher, Nancy, Josh McDermott, and Marvin M. Chun (1997). "The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face Perception". In: *Journal of Neuroscience* 17.11, pp. 4302–4311. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.17-11-04302.1997. eprint: https://www.jneurosci.org/content/17/11/4302.full.pdf. URL: https://www.jneurosci.org/content/17/11/4302.

Kaschube, Matthias, Michael Schnabel, Siegrid Löwel, David M. Coppola, Leonard E. White, and Fred Wolf (2010). "Universality in the Evolution of Orientation Columns in the Visual Cortex". In: *Science* 330.6007, pp. 1113–1116. DOI: 10.1126/science.1194869. eprint: https://www.science.org/doi/pdf/10.1126/science.1194869. URL: https://www.science.org/doi/abs/10.1126/science.1194869.

Kavukcuoglu, Koray, Marc'Aurelio Ranzato, Rob Fergus, and Yann LeCun (2009). "Learning invariant features through topographic filter maps". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1605–1612.

Keller, Georg B. and Thomas D. Mrsic-Flogel (2018). "Predictive Processing: A Canonical Cortical Computation". In: *Neuron* 100.2, pp. 424–435. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2018.10.003.

Keller, T. Anderson and Max Welling (2021a). "Topographic VAEs learn Equivariant Capsules". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. URL: https://openreview.net/forum?id=AVWROGUWpu.

Keller, T. Anderson and Max Welling (2021b). "Topographic VAEs learn Equivariant Capsules". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc.

Keurti, Hamza, Hsiao-Ru Pan, Michel Besserve, Benjamin F. Grewe, and Bernhard Schölkopf (2022). *Homomorphism Autoencoder – Learning Group Structured Representations from Observed Transitions*. DOI: 10.48550/ARXIV.2207.12067. URL: https://arxiv.org/abs/2207.12067.

Khosla, Meenakshi, N. Apurva Ratan Murty, and Nancy Kanwisher (2022). "A highly selective response to food in human visual cortex revealed by hypothesis-free voxel decomposition". In: *Current Biology* 32.19, 4159–4171.e9. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2022.08.009. URL: https://www.sciencedirect.com/science/article/pii/S0960982222012866.

Kietzmann, Tim C., Courtney J. Spoerer, Lynn K. A. Sörensen, Radoslaw M. Cichy, Olaf Hauk, and Nikolaus Kriegeskorte (2019). "Recurrence is required to capture the representational dynamics of the human visual system". In: *Proceedings of the National Academy of Sciences* 116.43, pp. 21854–21863. DOI: 10.1073/pnas.1905544116. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1905544116. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1905544116.

Kim, Hyunjik and Andriy Mnih (2018). "Disentangling by factorising". In: *ICML*.

King, Jean-Rémi and Valentin Wyart (Apr. 2021). "The Human Brain Encodes a Chronicle of Visual Events at Each Instant of Time Through the Multiplexing of Traveling Waves". In: *The Journal of Neuroscience* 41.34, pp. 7224–7233. DOI: 10.1523/jneurosci.2098-20.2021. URL: https://doi.org/10.1523/jneurosci.2098-20.2021.

Kingma, Diederik P. and Prafulla Dhariwal (2018). "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 10236–10245.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML].

Klindt, David, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton (2021). "Towards nonlinear disentanglement in natural data with temporal sparse coding". In: *ICLR*.

Klindt, David A., Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton (2021). "Towards Nonlinear Disentanglement in Natural Data with Temporal Sparse Coding". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=EbIDjBynYJ8.

Kohonen, Teuvo (1996). "Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map". In: *Biological cybernetics* 75.4, pp. 281–291.

Kolouri, Soheil, Navid Naderializadeh, Gustavo K Rohde, and Heiko Hoffmann (2021). "Wasserstein embedding for graph learning". In: *ICLR*.

Konkle, Talia and George A. Alvarez (2021). "Beyond category-supervision: Computational support for domain-general pressures guiding human visual system representation". In: *bioRxiv*. DOI: 10.1101/2020.06.15.153247. eprint: https://www.biorxiv.org/content/early/2021/10/19/2020.06.15.153247.full.pdf. URL: https://www.biorxiv.org/content/early/2021/10/19/2020.06.15.153247.

Konkle, Talia and Alfonso Caramazza (2013). "Tripartite Organization of the Ventral Stream by Animacy and Object Size". In: *Journal of Neuroscience* 33.25, pp. 10235–10242. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.0983-13.2013. eprint: https://www.jneurosci.org/content/33/25/10235.full.pdf. URL: https://www.jneurosci.org/content/33/25/10235.

Konkle, Talia and Aude Oliva (June 2012). "A Real-World Size Organization of Object Responses in Occipitotemporal Cortex". In: *Neuron* 74, pp. 1114–24. DOI: 10.1016/j.neuron.2012.04.036.

Koulakov, Alexei A and Dmitri B Chklovskii (2001). "Orientation preference patterns in mammalian visual cortex: a wire length minimization approach". In: *Neuron* 29.2, pp. 519–527.

Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton (n.d.). "CIFAR-10 (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Kubilius, Jonas, Martin Schrimpf, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, Aran Nayebi, Daniel Bear, Daniel L. K. Yamins, and James J. DiCarlo (2019). "Brain-Like Object Recognition with High-Performing Shallow Recurrent ANNs". In: *Neural Information Processing Systems (NeurIPS)*. Ed. by H. Wallach, H.

Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 12785–12796. URL: http://papers.nips.cc/paper/9441-brain-like-object-recognition-with-high-performing-shallow-recurrent-anns.

Kügelgen, Julius von, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello (2021). "Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 16451–16467. URL: https://proceedings.neurips.cc/paper/2021/file/8929c70f8d710e412d38da624b21c3c8-Paper.pdf.

Kumar, Abhishek, Prasanna Sattigeri, and Avinash Balakrishnan (2018). "Variational inference of disentangled latent concepts from unlabeled observations". In: *ICLR*.

Kuramoto, Yoshiki (1981). "Rhythms and turbulence in populations of chemical oscillators". In: *Physica A: Statistical Mechanics and its Applications* 106.1-2, pp. 128–143.

Kwon, Mingi, Jaeseok Jeong, and Youngjung Uh (2023). "Diffusion models already have a semantic latent space". In: *ICLR*.

Lake, Brenden M., Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman (2017). "Building machines that learn and think like people". In: *Behavioral and Brain Sciences* 40, e253. DOI: 10.1017/S0140525X16001837.

Le, Quoc V., Navdeep Jaitly, and Geoffrey E. Hinton (2015). *A Simple Way to Initialize Recurrent Networks of Rectified Linear Units*. arXiv: 1504.00941 [cs.NE].

Le, Ya and Xuan S. Yang (2015). "Tiny ImageNet Visual Recognition Challenge". In.

LeCun, Yann (1998). "The MNIST database of handwritten digits". In: URL: http://yann.lecun.com/exdb/mnist/.

LeCun, Yann and Corinna Cortes (2010). "MNIST handwritten digit database". In: URL: http://yann.lecun.com/exdb/mnist/.

LeCun, Yann, Corinna Cortes, and CJ Burges (2010). "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2.

LeCun, Yann and Ishan Misra (Mar. 2021). *Self-supervised learning: The dark matter of intelligence*. URL: https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/.

Lee, Chen-Yu, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu (2015). "Deeply-supervised nets". In: *AISTATS*. PMLR.

Lee, Dong-Hyun, Saizheng Zhang, Asja Fischer, and Yoshua Bengio (2015). "Difference Target Propagation". In: *Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*. ECMLPKDD'15. Porto, Portugal, pp. 498–515.

Lee, Hyodong, Eshed Margalit, Kamila M. Jozwik, Michael A. Cohen, Nancy Kanwisher, Daniel L. K. Yamins, and James J. DiCarlo (July 2020). "Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network". In: *bioRxiv*. DOI: https://doi.org/10.1101/2020.07.09.185116. URL: https://www.biorxiv.org/content/10.1101/2020.07.09.185116v1.full.pdf.

Lee, Jason D, Qi Lei, Nikunj Saunshi, and JIACHENG ZHUO (2021). "Predicting What You Already Know Helps: Provable Self-Supervised Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 309–323. URL: https://proceedings.neurips.cc/paper/2021/file/02e656adee09f8394b402d9958389b7d-Paper.pdf.

Lenc, Karel and Andrea Vedaldi (2015). "Understanding image representations by measuring their equivariance and equivalence". In: *CVPR*.

Lenssen, Jan Eric, Matthias Fey, and Pascal Libuschewski (2018). "Group Equivariant Capsule Networks". In: *NeurIPS*, pp. 8858–8867. URL: http://papers.nips.cc/paper/8100-group-equivariant-capsule-networks.

Lezcano-Casado, Mario and David Martínez-Rubio (2019). *Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group*. arXiv: 1901.08428 [cs.LG].

Li, Bo, Qili Wang, and Gim Hee Lee (18–24 Jul 2021). "FILTRA: Rethinking Steerable CNN by Filter Transform". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 6515–6522. URL: https://proceedings.mlr.press/v139/li21v.html.

Li, S., W. Li, C. Cook, C. Zhu, and Y. Gao (June 2018). "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 5457–5466. DOI: 10.1109/CVPR.2018.00572. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00572.

Lillicrap, Timothy P., Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman (2014). *Random feedback weights support learning in deep neural networks*. arXiv: 1411.0247 [q-bio.NC].

Lillicrap, Timothy P., Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton (Apr. 2020). "Backpropagation and the brain". In: *Nature Reviews Neuro-*

science 21.6, pp. 335–346. DOI: 10.1038/s41583-020-0277-3. URL: https://doi.org/10.1038/s41583-020-0277-3.

Locatello, Francesco, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen (2020). "Weakly-supervised disentanglement without compromises". In: *ICML*. PMLR.

Lotter, William, Gabriel Kreiman, and David Cox (2018). *A neural network trained to predict future video frames mimics critical properties of biological neuronal responses and perception*. arXiv: 1805.10734 [q-bio.NC].

Löwe, Sindy, Phillip Lippe, Maja Rudolph, and Max Welling (2022). *Complex-Valued Autoencoders for Object Discovery*. DOI: 10.48550/ARXIV.2204.02075. URL: https://arxiv.org/abs/2204.02075.

Löwe, Sindy, Peter O'Connor, and Bastiaan Veeling (2019). "Putting An End to End-to-End: Gradient-Isolated Learning of Representations". In: *Advances in Neural Information Processing Systems*.

Löwe, Sindy, Peter O'Connor, and Bastiaan Veeling (2019). "Putting An End to End-to-End: Gradient-Isolated Learning of Representations". In: *Advances in Neural Information Processing Systems 32*, pp. 3039–3051.

Lubenov, Evgueniy V. and Athanassios G. Siapas (May 2009). "Hippocampal theta oscillations are travelling waves". In: *Nature* 459.7246, pp. 534–539. DOI: 10.1038/nature08010. URL: https://doi.org/10.1038/nature08010.

Lukoševičius, Mantas and Herbert Jaeger (2009). "Reservoir computing approaches to recurrent neural network training". In: *Computer science review* 3.3, pp. 127–149.

Lyu, S and E P Simoncelli (June 2008). "Nonlinear image representation using divisive normalization". In: *Proc. Computer Vision and Pattern Recognition*. IEEE Computer Society, pp. 1–8. DOI: 10.1109/CVPR.2008.4587821.

Lyu, S and E P Simoncelli (Apr. 2009a). "Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures". In: *IEEE Trans. Patt. Analysis and Machine Intelligence* 31.4, pp. 693–706. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.107.

Lyu, S and E P Simoncelli (Apr. 2009b). "Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures". In: *IEEE Trans. Patt. Analysis and Machine Intelligence* 31.4, pp. 693–706. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.107.

Ma, Libo and Liqing Zhang (2008). "Overcomplete topographic independent component analysis". In: *Neurocomputing* 71.10-12, pp. 2217–2223.

Marino, Joseph (2020). *Predictive Coding, Variational Autoencoders, and Biological Connections*. arXiv: 2011.07464 [cs.NE].

Matthey, Loic, Irina Higgins, Demis Hassabis, and Alexander Lerchner (2017). *dSprites: Disentanglement testing Sprites dataset*. URL: https://github.com/deepmind/dsprites-dataset/.

McCulloch, Warren S. and Walter Pitts (Dec. 1943). "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5.4, pp. 115–133. DOI: 10.1007/bf02478259. URL: https://doi.org/10.1007/bf02478259.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (Dec. 2021). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Commun. ACM* 65.1, pp. 99–106. ISSN: 0001-0782. DOI: 10.1145/3503250. URL: https://doi.org/10.1145/3503250.

Moravec, Hans P (July 1990). *Mind children*. London, England: Harvard University Press.

Moro, Valentina, Cosimo Urgesi, Simone Pernigo, Paola Lanteri, Mariella Pazzaglia, and Salvatore Aglioti (Nov. 2008). "The Neural Basis of Body Form and Body Action Agnosia". In: *Neuron* 60, pp. 235–46. DOI: 10.1016/j.neuron.2008.09.022.

Muller, Lyle, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski (Mar. 2018). "Cortical travelling waves: mechanisms and computational principles". In: *Nature Reviews Neuroscience* 19.5, pp. 255–268. DOI: 10.1038/nrn.2018.20. URL: https://doi.org/10.1038/nrn.2018.20.

Muller, Lyle, Giovanni Piantoni, Dominik Koller, Sydney S Cash, Eric Halgren, and Terrence J Sejnowski (Nov. 2016). "Rotating waves during human sleep spindles organize global patterns of activity that repeat precisely through the night". In: *eLife* 5. Ed. by Frances K Skinner, e17267. ISSN: 2050-084X. DOI: 10.7554/eLife.17267. URL: https://doi.org/10.7554/eLife.17267.

Muller, Lyle, Alexandre Reynaud, Frédéric Chavane, and Alain Destexhe (Apr. 2014). "The stimulus-evoked population response in visual cortex of awake monkey is a propagating wave". In: *Nature Communications* 5.1. DOI: 10.1038/ncomms4675. URL: https://doi.org/10.1038/ncomms4675.

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, pp. 807–814. ISBN: 9781605589077.

Nasr, Shahin, Ning Liu, Kathryn J. Devaney, Xiaomin Yue, Reza Rajimehr, Leslie G. Ungerleider, and Roger B. H. Tootell (2011). "Scene-Selective Cortical Regions in Human and Nonhuman Primates". In: *Journal of Neuroscience* 31.39, pp. 13771–13785. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.2792-11.2011. eprint: https://www.jneurosci.org/content/31/39/13771.full.pdf. URL: https://www.jneurosci.org/content/31/39/13771.

Neftci, Emre O., Hesham Mostafa, and Friedemann Zenke (2019). *Surrogate Gradient Learning in Spiking Neural Networks*. DOI: 10.48550/ARXIV.1901.09948. URL: https://arxiv.org/abs/1901.09948.

Neklyudov, Kirill, Daniel Severo, and Alireza Makhzani (2023). "Action Matching: A Variational Method for Learning Stochastic Dynamics from Samples". In: *ICML*.

Neumann, J. von (1945). "First draft of a report on the EDVAC". In: *IEEE Annals of the History of Computing* 15.4. Republished in IEEE Annals in 1993, pp. 27–75. DOI: 10.1109/85.238389.

Oldfield, James, Christos Tzelepis, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras (2023). "PandA: Unsupervised Learning of Parts and Appearances in the Feature Maps of GANs". In: *ICLR*.

Olshausen, Bruno A and David J Field (1997). "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision research* 37.23, pp. 3311–3325.

Onken, Derek, Samy Wu Fung, Xingjian Li, and Lars Ruthotto (2021). "Ot-flow: Fast and accurate continuous normalizing flows via optimal transport". In: *AAAI*.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). *Representation Learning with Contrastive Predictive Coding*. DOI: 10.48550/ARXIV.1807.03748. URL: https://arxiv.org/abs/1807.03748.

Orvieto, Antonio, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De (2023). *Resurrecting Recurrent Neural Networks for Long Sequences*. arXiv: 2303.06349 [cs.LG].

Osindero, Simon, Max Welling, and Geoffrey E. Hinton (Feb. 2006). "Topographic Product Models Applied to Natural Scene Statistics". In: *Neural Computation* 18.2, pp. 381–414. ISSN: 0899-7667. DOI: 10.1162/089976606775093936. eprint: https://direct.mit.edu/neco/article-pdf/18/2/381/816474/089976606775093936.pdf. URL: https://doi.org/10.1162/089976606775093936.

Osindero, Simon Kayode (2004). "Contrastive Topographic Models". PhD thesis. University of London.

Ouden, Hanneke E. M. den, Jean Daunizeau, Jonathan Roiser, Karl J. Friston, and Klaas E. Stephan (2010). "Striatal Prediction Error Modulates Cortical Coupling". In: *Journal of Neuroscience* 30.9, pp. 3210–3219. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.4458-09.2010. eprint: https://www.jneurosci.org/content/30/9/3210.full.pdf. URL: https://www.jneurosci.org/content/30/9/3210.

Pal, Dipan K. and Marios Savvides (2018). *Non-Parametric Transformation Networks*. DOI: 10.48550/ARXIV.1801.04520. URL: https://arxiv.org/abs/1801.04520.

Park, Yong-Hyun, Mingi Kwon, Junghyo Jo, and Youngjung Uh (2023). "Unsupervised Discovery of Semantic Latent Directions in Diffusion Models". In: *arXiv preprint arXiv:2302.12469*.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*, pp. 8024–8035.

Patrini, Giorgio, Rianne van den Berg, Patrick Forre, Marcello Carioni, Samarth Bhargav, Max Welling, Tim Genewein, and Frank Nielsen (2020). "Sinkhorn autoencoders". In: *UAI*.

Peebles, William, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba (2020). "The hessian penalty: A weak prior for unsupervised disentanglement". In: *ECCV*.

Peelen, Marius V. and Paul E. Downing (2005). "Selectivity for the Human Body in the Fusiform Gyrus". In: *Journal of Neurophysiology* 93.1. PMID: 15295012, pp. 603–608. DOI: 10.1152/jn.00513.2004. eprint: https://doi.org/10.1152/jn.00513.2004. URL: https://doi.org/10.1152/jn.00513.2004.

Perrard, Stéphane, Emmanuel Fort, and Yves Couder (2016). "Wave-based turing machine: Time reversal and information erasing". In: *Physical review letters* 117.9, p. 094502.

Pinker, Steven (Sept. 2007). *The language instinct*. New York, NY: HarperCollins.

Pinsk, Mark A., Kevin DeSimone, Tirin Moore, Charles G. Gross, and Sabine Kastner (2005a). "Representations of faces and body parts in macaque temporal cortex: A functional MRI study". In: *Proceedings of the National Academy of Sciences* 102.19, pp. 6996–7001. ISSN: 0027-8424. DOI: 10.1073/pnas.0502605102. eprint: https://www.pnas.org/content/102/19/6996.full.pdf. URL: https://www.pnas.org/content/102/19/6996.

Pinsk, Mark A., Kevin DeSimone, Tirin Moore, Charles G. Gross, and Sabine Kastner (2005b). "Representations of faces and body parts in macaque temporal cortex: A functional MRI study". In: *Proceedings of the National Academy of Sciences* 102.19, pp. 6996–7001. ISSN: 0027-8424. DOI: 10.1073/pnas.0502605102. eprint: https://www.pnas.org/content/102/19/6996.full.pdf. URL: https://www.pnas.org/content/102/19/6996.

Pol, Elise van der, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling (2020). "MDP Homomorphic Networks: Group Symmetries in Reinforcement Learning". In: *CoRR* abs/2006.16908. arXiv: 2006.16908. URL: https://arxiv.org/abs/2006.16908.

Portilla, J, V Strela, M J Wainwright, and E P Simoncelli (Nov. 2003). "Image denoising using scale mixtures of Gaussians in the wavelet domain". In: *IEEE Trans Image Processing* 12.11. Recipient, IEEE Signal Processing Society Best Paper Award, 2008., pp. 1338–1351. DOI: 10.1109/TIP.2003.818640.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). *Learning Transferable Visual Models From Natural Language Supervision*. DOI: 10.48550/ARXIV.2103.00020. URL: https://arxiv.org/abs/2103.00020.

Radford, Alec, Luke Metz, and Soumith Chintala (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv: 1511.06434 [cs.LG].

Raissi, M., P. Perdikaris, and G.E. Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics*.

Raman, Rajani and Haruo Hosoya (2020). "Convolutional neural networks explain tuning properties of anterior, but not middle, face-processing areas in macaque inferotemporal cortex". In: *Communications Biology* 3.1, p. 221. DOI: 10.1038/s42003-020-0945-x. URL: https://doi.org/10.1038/s42003-020-0945-x.

Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen (2022). *Hierarchical Text-Conditional Image Generation with CLIP Latents*. DOI: 10.48550/ARXIV.2204.06125. URL: https://arxiv.org/abs/2204.06125.

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). *Zero-Shot Text-to-Image Generation*. DOI: 10.48550/ARXIV.2102.12092. URL: https://arxiv.org/abs/2102.12092.

Rao, Rajesh and Dana Ballard (Feb. 1999). "Predictive Coding in the Visual Cortex: a Functional Interpretation of Some Extra-classical Receptive-field Effects". In: *Nature neuroscience* 2, pp. 79–87. DOI: 10.1038/4580.

Ravanbakhsh, Siamak, Jeff Schneider, and Barnabás Póczos (June 2017). "Equivariance Through Parameter-Sharing". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 2892–2901. URL: https://proceedings.mlr.press/v70/ravanbakhsh17a.html.

Rezende, Danilo and Shakir Mohamed (2015). "Variational inference with normalizing flows". In: *ICML*. PMLR.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, II–1278–II–1286.

Ricci, Matthew, Minju Jung, Yuwei Zhang, Mathieu Chalvidal, Aneri Soni, and Thomas Serre (2021). *KuraNet: Systems of Coupled Oscillators that Learn to Synchronize*. DOI: 10.48550/ARXIV.2105.02838. URL: https://arxiv.org/abs/2105.02838.

Richter-Powell, Jack, Yaron Lipman, and Ricky TQ Chen (2022). "Neural conservation laws: A divergence-free perspective". In: *NeurIPS*.

Ridgeway, Karl and Michael C Mozer (2018). "Learning deep disentangled embeddings with the f-statistic loss". In: *NeurIPS*.

Romero, David W., Robert-Jan Bruintjes, Jakub Mikolaj Tomczak, Erik J Bekkers, Mark Hoogendoorn, and Jan van Gemert (2022). "FlexConv: Continuous Kernel Convolutions With Differentiable Kernel Sizes". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=3jooF27-0Wy.

Rosenblatt, F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6, pp. 386–408. DOI: 10.1037/h0042519. URL: https://doi.org/10.1037/h0042519.

Rusch, T Konstantin, Siddhartha Mishra, N Benjamin Erichson, and Michael W Mahoney (2022). "Long Expressive Memory for Sequence Modeling". In: *International Conference on Learning Representations*.

Rusch, T. Konstantin and Siddhartha Mishra (2021a). "Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies". In: *International Conference on Learning Representations*.

Rusch, T. Konstantin and Siddhartha Mishra (2021b). *UnICORNN: A recurrent model for learning very long time dependencies*. DOI: 10.48550/ARXIV.2103.05487. URL: https://arxiv.org/abs/2103.05487.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton (2017). "Dynamic Routing between Capsules". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., pp. 3859–3869. ISBN: 9781510860964.

Salimans, Tim, Han Zhang, Alec Radford, and Dimitris Metaxas (2018). "Improving GANs using optimal transport". In: *ICLR*.

Sandfort, Veit, Ke Yan, Perry J. Pickhardt, and Ronald M. Summers (Nov. 2019). "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks". In: *Scientific Reports* 9.1. DOI: 10.1038/s41598-019-52737-x. URL: https://doi.org/10.1038/s41598-019-52737-x.

Santoro, Adam, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap (2016). *One-shot Learning with Memory-Augmented Neural Networks*. arXiv: 1605.06065 [cs.LG].

Sato, Naoyuki (Jan. 2022). "Cortical traveling waves reflect state-dependent hierarchical sequencing of local regions in the human connectome network". In: *Scientific Reports* 12.1. DOI: 10.1038/s41598-021-04169-9. URL: https://doi.org/10.1038/s41598-021-04169-9.

Sato, Tatsuo K., Ian Nauhaus, and Matteo Carandini (July 2012). "Traveling Waves in Visual Cortex". In: *Neuron* 75.2, pp. 218–229. DOI: 10.1016/j.neuron.2012.06.029. URL: https://doi.org/10.1016/j.neuron.2012.06.029.

Sawilowsky, Shlomo S. (Nov. 2009). "New Effect Size Rules of Thumb". In: *Journal of Modern Applied Statistical Methods* 8.2, pp. 597–599. DOI: 10.22237/jmasm/1257035100. URL: http://dx.doi.org/10.22237/jmasm/1257035100.

Saygin, Zeynep M., David E. Osher, Kami Koldewyn, Gretchen O Reynolds, John D. E. Gabrieli, and Rebecca Saxe (2012). "Anatomical connectivity patterns predict face-selectivity in the fusiform gyrus". In: *Nature neuroscience* 15, pp. 321–327.

Schaul, Tom, John Quan, Ioannis Antonoglou, and David Silver (2016). *Prioritized Experience Replay*. arXiv: 1511.05952 [cs.LG].

Schmidhuber, Jürgen (1992). "Learning factorial codes by predictability minimization". In: *Neural computation*.

Schmidt, Uwe and Stefan Roth (2012). "Learning rotation-aware features: From invariant priors to equivariant descriptors". In: *CVPR*.

Schreiner, Maximilian (July 2023). *GPT-4 architecture, datasets, costs and more leaked*. URL: https://the-decoder.com/gpt-4-architecture-datasets-costs-and-more-leaked/.

Serre, Jean-Pierre (1977). *Linear representations of finite groups*. Vol. 42. Graduate texts in mathematics. Springer, pp. I–X, 1–170. ISBN: 978-3-540-90190-7.

Shen, Yujun and Bolei Zhou (2021). "Closed-form factorization of latent semantics in gans". In: *CVPR*.

Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. arXiv: 1506.04214 [cs.CV].

Shi, Yuge, N Siddharth, Philip HS Torr, and Adam R Kosiorek (2022). "Adversarial Masking for Self-Supervised Learning". In: *International Conference on Machine Learning*.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis (Jan. 2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961. URL: https://doi.org/10.1038/nature16961.

Simoncelli, E.P. and W.T. Freeman (n.d.). "The steerable pyramid: a flexible architecture for multi-scale derivative computation". In: *Proceedings., International Conference on Image Processing*. IEEE Comput. Soc. Press. DOI: 10.1109/icip.1995.537667. URL: https://doi.org/10.1109/icip.1995.537667.

Sitzmann, Vincent, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein (2020). "Implicit Neural Representations with Periodic Activation Functions". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc. ISBN: 9781713829546.

Song, Yue, T. Anderson Keller, Nicu Sebe, and Max Welling (2023). "Latent Traversals in Generative Models as Potential Flows". In: *ICML*. PMLR.

Song, Yue, Nicu Sebe, and Wei Wang (2022). "Orthogonal SVD Covariance Conditioning and Latent Disentanglement". In: *IEEE T-PAMI*.

Sosnovik, Ivan, Michał Szmaja, and Arnold Smeulders (2020). "Scale-Equivariant Steerable Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HJgpugrKPS.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Steinmetz, Nicholas A., Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, Susu Chen, Jennifer Colonell, Richard J. Gardner, Bill Karsh, Fabian Kloosterman, Dimitar Kostadinov, Carolina Mora-Lopez, John O'Callaghan, Junchol Park, Jan Putzeys, Britton Sauerbrei, Rik J. J. van Daal, Abraham Z. Vollan, Shiwei Wang, Marleen Welkenhuysen, Zhiwen Ye, Joshua T. Dudman, Barundeb Dutta, Adam W. Hantman, Kenneth D. Harris, Albert K. Lee, Edvard I.

Moser, John O'Keefe, Alfonso Renart, Karel Svoboda, Michael Häusser, Sebastian Haesler, Matteo Carandini, and Timothy D. Harris (Apr. 2021). "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings". In: *Science* 372.6539. DOI: 10.1126/science.abf4588. URL: https://doi.org/10.1126/science.abf4588.

Stone, James V. (Oct. 1996). "Learning Perceptually Salient Visual Parameters Using Spatiotemporal Smoothness Constraints". In: *Neural Computation* 8.7, pp. 1463–1492. ISSN: 0899-7667. DOI: 10.1162/neco.1996.8.7.1463. eprint: https://direct.mit.edu/neco/article-pdf/8/7/1463/813515/neco.1996.8.7.1463.pdf. URL: https://doi.org/10.1162/neco.1996.8.7.1463.

Stühmer, Jan, Richard E. Turner, and Sebastian Nowozin (2019). *Independent Subspace Analysis for Unsupervised Learning of Disentangled Representations*. arXiv: 1909.05063 [stat.ML].

Swindale, N. V. (1982). "A Model for the Formation of Orientation Columns". In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 215.1199, pp. 211–230. ISSN: 00804649. URL: http://www.jstor.org/stable/35596 (visited on 11/22/2022).

Tai, Chang-Yu, Ming-Yao Li, and Lun-Wei Ku (2022). "Hyperbolic disentangled representation for fine-grained aspect extraction". In: *AAAI*.

Tallec, Corentin and Yann Ollivier (2018). *Can recurrent neural networks warp time?* arXiv: 1804.11188 [cs.LG].

Tenenbaum, Joshua B. (1999). *A Bayesian framework for concept learning*.

Thoppilan, Romal, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le (2022). *LaMDA: Language Models for Dialog Applications*. arXiv: 2201.08239 [cs.CL].

Tian, Yonglong, Dilip Krishnan, and Phillip Isola (2019). *Contrastive Multiview Coding*. DOI: 10.48550/ARXIV.1906.05849. URL: https://arxiv.org/abs/1906.05849.

Tian, Yonglong, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola (2020). "What Makes for Good Views for Contrastive Learning?" In: *arXiv preprint arXiv:2005.10243*.

Tian, Yuandong, Xinlei Chen, and Surya Ganguli (2021). "Understanding self-supervised Learning Dynamics without Contrastive Pairs". In: *CoRR* abs/2102.06810. arXiv: 2102.06810. URL: https://arxiv.org/abs/2102.06810.

Tian, Yuandong, Lantao Yu, Xinlei Chen, and Surya Ganguli (2020). *Understanding Self-supervised Learning with Dual Deep Networks*. DOI: 10.48550/ARXIV.2010.00578. URL: https://arxiv.org/abs/2010.00578.

Tolstikhin, Ilya, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf (2018). "Wasserstein auto-encoders". In: *ICLR*.

Tomczak, Jakub and Max Welling (2018). "VAE with a VampPrior". In: *AISTATS*. PMLR.

Tong, Alexander, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy (2020). "Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics". In: *ICML*. PMLR.

Tosh, Christopher, Akshay Krishnamurthy, and Daniel Hsu (2020). *Contrastive learning, multi-view redundancy, and linear models*. DOI: 10.48550/ARXIV.2008.10150. URL: https://arxiv.org/abs/2008.10150.

Tsai, Yao-Hung Hubert, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov (2022). "Conditional Contrastive Learning with Kernel". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=AAJLBoGt0XM.

Tsai, Yao-Hung Hubert, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency (2020). *Self-supervised Learning from a Multi-view Perspective*. DOI: 10.48550/ARXIV.2006.05576. URL: https://arxiv.org/abs/2006.05576.

Tsao, Doris Y., Winrich A. Freiwald, Roger B. H. Tootell, and Margaret S. Livingstone (2006). "A cortical region consisting entirely of face-selective cells." In: *Science* 311.5761, pp. 670–674.

Turing, Alan (Sept. 2004). "Intelligent Machinery (1948)". In: *The Essential Turing*. Oxford University Press. ISBN: 9780198250791. DOI: 10.1093/oso/9780198250791.003.0016. eprint: https://academic.oup.com/book/0/chapter/355746030/chapter-pdf/43817005/isbn-9780198250791-book-part-16.pdf. URL: https://doi.org/10.1093/oso/9780198250791.003.0016.

Tzelepis, Christos, Georgios Tzimiropoulos, and Ioannis Patras (2021). "WarpedGANSpace: Finding Non-Linear RBF Paths in GAN Latent Space". In: *ICCV*.

Vahdat, Arash and Jan Kautz (2020). "NVAE: A Deep Hierarchical Variational Autoencoder". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 19667–19679. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

Van der Pol, Elise, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling (2020). "Mdp homomorphic networks: Group symmetries in reinforcement learning". In: *NeurIPS*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *NeurIPS*.

Veeling, Bastiaan S., Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling (2018). "Rotation Equivariant CNNs for Digital Pathology". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger. Cham: Springer International Publishing, pp. 210–218. ISBN: 978-3-030-00934-2.

Verma, Vikas, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio (2018). *Manifold Mixup: Better Representations by Interpolating Hidden States*. DOI: 10.48550/ARXIV.1806.05236. URL: https://arxiv.org/abs/1806.05236.

Villani, Cédric (2009). *Optimal transport: old and new*. Vol. 338. Springer.

Villani, Cédric (2021). *Topics in optimal transportation*. Vol. 58. American Mathematical Soc.

Voynov, Andrey and Artem Babenko (2020). "Unsupervised discovery of interpretable directions in the gan latent space". In: *ICML*.

Wainwright, M J and E P Simoncelli (May 2000). "Scale mixtures of Gaussians and the statistics of natural images". In: *Adv. Neural Information Processing Systems (NIPS*99)*. Ed. by S. A. Solla, T. K. Leen, and K.-R. Müller. Vol. 12. Cambridge, MA: MIT Press, pp. 855–861.

Wainwright, M J, E P Simoncelli, and A S Willsky (July 2001). "Random cascades on wavelet trees and their use in analyzing and modeling natural images". In: *Applied and Computational Harmonic Analysis* 11.1, pp. 89–123. DOI: 10.1117/12.408598.

Wang, Yifei, Zhengyang Geng, Feng Jiang, Chuming Li, Yisen Wang, Jiansheng Yang, and Zhouchen Lin (2021). "Residual Relaxation for Multi-view Representation Learning". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. URL: https://openreview.net/forum?id=rEBScZF6G70.

Wang, Yifei, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin (2022). *Chaos is a Ladder: A New Theoretical Understanding of Contrastive Learning via Augmentation Overlap*. arXiv: 2203.13457 [cs.LG].

Wei, Yuxiang, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo (2021). "Orthogonal jacobian regularization for unsupervised disentanglement in image generation". In: *ICCV*.

Weiler, Maurice and Gabriele Cesa (2019). "General E(2)-Equivariant Steerable CNNs". In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Weiler, Maurice, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen (2018). "3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., pp. 10402–10413.

Weiner, Kevin and Kalanit Grill-Spector (Dec. 2011). "Neural representations of faces and limbs neighbor in human high-level visual cortex: Evidence for a new organization principle". In: *Psychological research* 77. DOI: 10.1007/s00426-011-0392-x.

Weiner, Kevin S., Golijeh Golarai, Julian Caspers, Miguel R. Chuapoco, Hartmut Mohlberg, Karl Zilles, Katrin Amunts, and Kalanit Grill-Spector (2014). "The mid-fusiform sulcus: A landmark identifying both cytoarchitectonic and functional divisions of human ventral temporal cortex". In: *NeuroImage* 84, pp. 453–465.

Welling, Max, Simon Osindero, and Geoffrey E Hinton (2003). "Learning sparse topographic representations with products of student-t distributions". In: *Advances in neural information processing systems*, pp. 1383–1390.

Whittington, James C.R., Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E.J. Behrens (2020). "The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation". In: *Cell* 183.5, 1249–1263.e23. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2020.10.024. URL: https://www.sciencedirect.com/science/article/pii/S009286742031388X.

Wiesel, Torsten N. and David H. Hubel (1974). "Ordered arrangement of orientation columns in monkeys lacking visual experience". In: *Journal of Comparative Neurology* 158.3, pp. 307–318. DOI: https://doi.org/10.1002/cne.901580306. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.901580306. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.901580306.

Wiskott, Laurenz and Terrence J Sejnowski (2002). "Slow feature analysis: Unsupervised learning of invariances". In: *Neural computation* 14.4, pp. 715–770.

Wolpert, David (Mar. 1996). "The Lack of A Priori Distinctions Between Learning Algorithms". In: *Neural Computation* 8. DOI: 10.1162/neco.1996.8.7.1341.

Worrall, Daniel and Max Welling (2019a). "Deep Scale-spaces: Equivariance Over Scale". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/f04cd7399b2b0128970efb6d20b5c551-Paper.pdf.

Worrall, Daniel and Max Welling (2019b). "Deep scale-spaces: Equivariance over scale". In: *NeurIPS*.

Worrall, Daniel E., Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow (2017). "Harmonic Networks: Deep Translation and Rotation Equivariance". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7168–7177.

Xiao, Tete, Xiaolong Wang, Alexei A Efros, and Trevor Darrell (2021). "What Should Not Be Contrastive in Contrastive Learning". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=CZ8Y3NzuVzO.

Yamins, Daniel L K and James J DiCarlo (2016). "Using goal-driven deep learning models to understand sensory cortex". In: *Nature Neuroscience* 19.3, pp. 356–365. DOI: 10.1038/nn.4244. URL: https://doi.org/10.1038/nn.4244.

Yamins, Daniel L. K., Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo (2014). "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the National Academy of Sciences* 111.23, pp. 8619–8624. DOI: 10.1073/pnas.1403112111. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1403112111. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1403112111.

Yang, Liu and George Em Karniadakis (2020). "Potential flow generator with L 2 optimal transport regularity for generative models". In: *IEEE TNNLS*.

Yang, Tao, Yuwang Wang, Yan Lv, and Nanning Zh (2023). "DisDiff: Unsupervised Disentanglement of Diffusion Probabilistic Models". In: *arXiv preprint arXiv:2301.13721*.

You, Yang, Igor Gitman, and Boris Ginsburg (2017). *Large Batch Training of Convolutional Networks*. DOI: 10.48550/ARXIV.1708.03888. URL: https://arxiv.org/abs/1708.03888.

Zanos, Theodoros P, Patrick J Mineault, Konstantinos T Nasiotis, Daniel Guitton, and Christopher C Pack (2015). "A sensorimotor role for traveling waves in primate visual cortex". In: *Neuron* 85.3, pp. 615–627.

Zeng, Qi, Spencer H Bryngelson, and Florian Schäfer (2023). "Competitive Physics Informed Networks". In: *ICLR*.

Zhang, Honghui, Andrew J. Watrous, Ansh Patel, and Joshua Jacobs (June 2018). "Theta and Alpha Oscillations Are Traveling Waves in the Human Neocortex". In: *Neuron* 98.6, 1269–1281.e4. DOI: 10.1016/j.neuron.2018.05.019. URL: https://doi.org/10.1016/j.neuron.2018.05.019.

Zhang, Richard (2019). "Making convolutional networks shift-invariant again". In: *International conference on machine learning*. PMLR, pp. 7324–7334.

Zhang, Yiyuan, Ke Zhou, Pinglei Bao, and Jia Liu (2021). "Principles governing the topological organization of object selectivities in ventral temporal cortex". In: *bioRxiv*. DOI: 10.1101/2021.09.15.460220. eprint: https://www.biorxiv.org/content/early/2021/09/17/2021.09.15.460220.full.pdf. URL: https://www.biorxiv.org/content/early/2021/09/17/2021.09.15.460220.

Zhou, Bolei, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba (2017). "Places: A 10 million Image Database for Scene Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhu, Jiapeng, Ruili Feng, Yujun Shen, Deli Zhao, Zheng-Jun Zha, Jingren Zhou, and Qifeng Chen (2021). "Low-rank subspaces in gans". In: *NeurIPS*.

Zhu, Jiapeng, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen (2022). "Region-Based Semantic Factorization in GANs". In: *ICML*.

Zhu, Xinqi, Chang Xu, and Dacheng Tao (2020). "Learning disentangled representations with latent variation predictability". In: *ECCV*.

Appendices

Appendix A

CHAPTER IV APPENDIX

A.1. Experiment Details – MNIST

Code for reproducing the 2D topographical organization of MNIST digits in Figure 4.1, as well as the general implementation of the topographic VAE can be found at: `https://github.com/AKAndyKeller/TopographicVAE`.

Optimizer Parameters

The 2D Topographic VAE without Temporal Coherence presented in Figure 4.1 was trained with stochastic gradient descent on batches of size 128, using a learning rate of 1×10^{-4}, and standard momentum of 0.9 for 250 epochs.

Initalization

All weights of the models were initialized with uniformly random samples from $U(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$, where m is the number of input units. For the 2D topographic model in Figure 4.1, μ was initialized to 10.

Model Architectures

All models presented in this paper make use of the same 3-Layer MLP for parameterizing the encoders and decoders. Specifically, the model is constructed as 3 fully connected layers with ReLU activations in-between the layers. For MNIST, the layers of both the $\mathbf{u}$ and $\mathbf{z}$ encoders have (972, 648, 648) output units each for the first, second, and third layers respectively. The 648 units in the third layer are divided into two sets to compute the mean and log standard deviation of the respective u's and z's, yielding 324 t variables. These variables were then arranges in a 18×18 square grid as shown in the figure.

Choice of $\mathbf{W}$

The Topographic VAE organizes its variables as a single 2-D torus. Practically, multiplication by $\mathbf{W}$ was performed by convolution over the appropriate dimensions

(time & capsule dimension) with a kernel of all 1's, taking advantage of circular padding to achieve toroidal structure.

A.2. Experiment Details – ImageNet

Code for reproducing the category selectivity experiments can be found at: `https://github.com/akandykeller/CategorySelectiveTVAE`.

Training details

Dataset Preprocessing — In order to eliminate variability between different datasets, all images were first reshaped to 256×256. A random percentage of the image area (between 8% to 100%) and a random aspect ratio (between $\frac{3}{4}$ and $\frac{4}{3}$) were then chosen, and each image was then cropped according to these values. Finally, the crops were resized to the final shape of 224×224. All images were then normalized by the mean $[0.48300076, 0.45126104, 0.3998704]$ and standard deviation $[0.26990137, 0.26078254, 0.27288908]$.

TDANN Hyperparameters — The TDANN model was trained with stochastic gradient descent, a learning rate of 1×10^{-3}, standard momentum of 0.9, and a batch size of 128 for 10 epochs. Explicitly, the loss function was given by a sum of the classification cross entropy loss, the spatial correlation losses for both layers FC6 and FC7, and weight decay of $5{\times}10^{-4}$. A fixed weight of $10{\times}\frac{1}{4096^2}$ was multiplied by the spatial correlation loss before backpropagating as this was found necessary to qualitatively match the results from H. Lee et al. (2020). Contrary to the original TDANN work, we did not randomly initialize the locations of the neurons, and instead spaced them evenly on a grid of the same size. We found the spatial correlation loss to still function equally well in this setting, and detail our implementation in Section **??** below.

TVAE Hyperparameters — The TVAE was trained with stochastic gradient descent, a learning rate of 1×10^{-5}, standard momentum of 0.9, and a batch size of 128 for 30 epochs. The global topology was set to a single 2D torus (i.e. a 2D grid with circular boundary conditions), and the local topology was set to sum of local regions of size 25×25, i.e. the kernel used to convolve over $\mathbf{u}$ was of size 25×25 and contained all 1's. The μ parameter was initialized to 40, and trained simultanously with the remainder of the model parameters.

A.3. Additional Results

Robustness to Initialization

To verify the robustness of our results to randomness between trials, in Figure A.1 below we compare the selectivity maps shown in the main text across four independant random initalizations of the weights. We first note that the emergent feature hierarchy depicted in Figure 4.5 appears roughly consistent across each trial. Specifically, selectivity to places, 'big', and 'inanimate' objects appears highly overlapping in each setting. We further note that the relative placement and size of the category-selective clusters (shown in the bottom row) is again roughly consistent across runs, with face and body clusters always adjacent and frequently overlapping. We see that in some runs, a small cluster selective to a generic 'object' category can be observed. The relative weakness of this cluster is likely due to the lack of uniquely identifying features shared across all images in the object dataset.

Robustness to Face Test-Dataset Choice

To investigate the robustness of face selectivity across different face test-datasets, and ensure the observed clusters are not a dataset dependant phenomenon, selectivity maps computed using four different face test-datasets are shown for both the TVAE and TDANN in Figure A.2 below. Explicitly, the four datasets included: a 25,000 subset of VGGface2 cao2018vggface2, 10,137 images from UTKface zhifei2017cvpr, 24,684 images from CelebA liu2015faceattributes, and the Labled Faces in the Wild Huang et al., 2007 dataset upon which the models were trained. The resulting selectivity maps can be seen to be highly consistent despite the variability between low-level dataset statistics, indicating the observed selectivity is more likely related to the high level category information as desired.

Distance-dependant Pairwise Correlation

To further quantify the topographic organization of the TVAE and how it compares with that of the TDANN, we measure the pairwise correlation (Pearson's R) of all topographic neurons as a function of distance in Figure A.3. We see that the TDANN (right) curve matches the original results H. Lee et al., 2020, roughly achieving the minimal spatial correlation loss, and mimicking the observed correlation curve from recordings in monkeys, as designed (see H. Lee et al., 2020 for further discussion). Interestingly, the TVAE (middle) yields a qualitatively similar curve, despite having

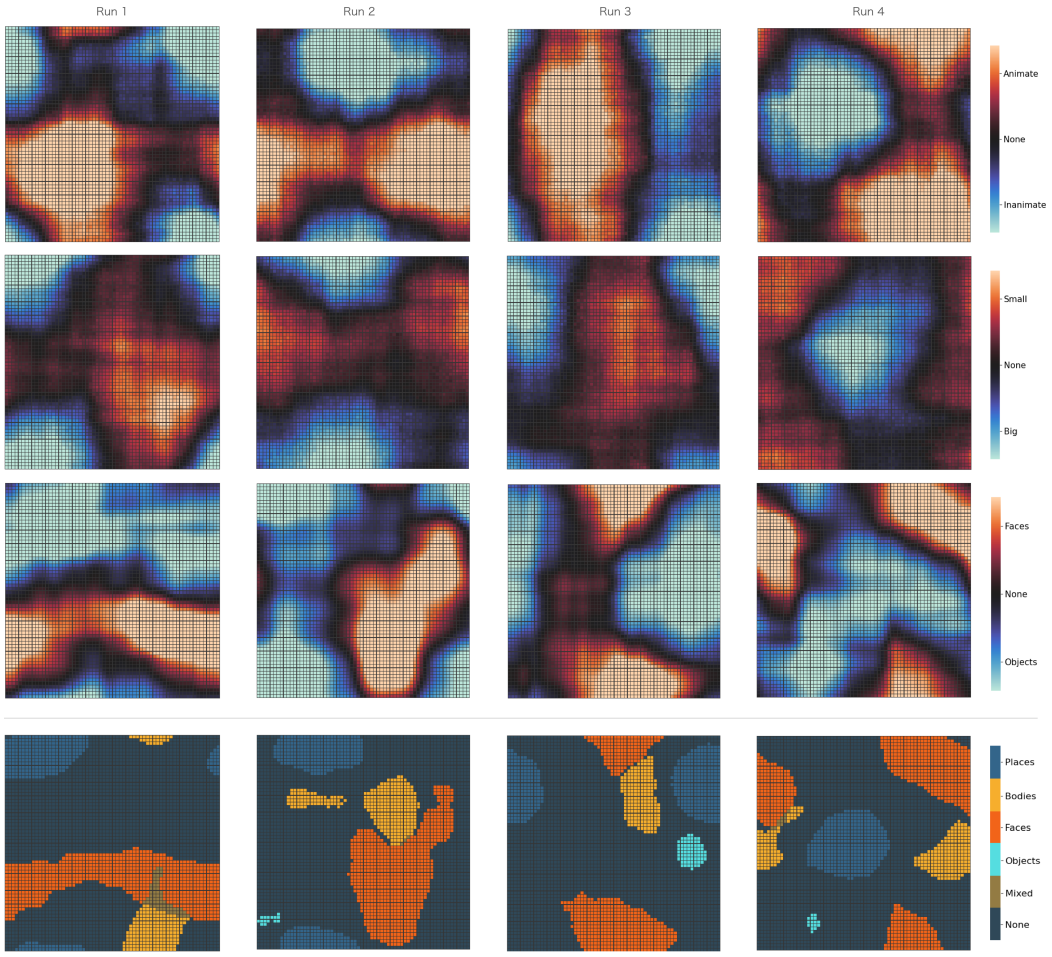


Figure A.1: Selectivity maps for the TVAE across four random initalizations. We observe that the emergent feature hierarchy and the relative placement of category-clusters is consistent in each case.

no such goal in its initial design. Finally, the correlation of the baseline model (left) is independant of distance as expected. We note that due to the circular boundary conditions of the TVAE, the maximal distance between neurons is significantly less, and thus scale of the X-axis is different between these two plots. In future work a more detailed comparison would benefit from matching boundary conditions in both models. Finally, in Figure A.4 we plot the correlation curves for TVAEs trained with different spatial window sizes. We see that this has a significant effect on the shape of the curve, potentially allowing for more precise tuning to match biological data.

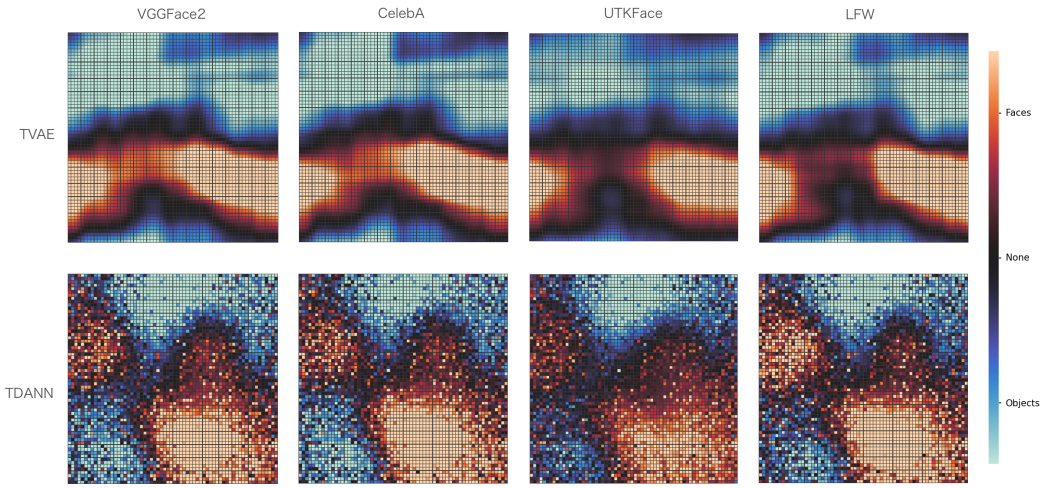


Figure A.2: Face vs. Object selectivity maps for four different face datasets. We see that for both the TVAE and TDANN the relative locations and sizes of the face and object selective clusters are stable despite the differences in the underlying test-datasets used.

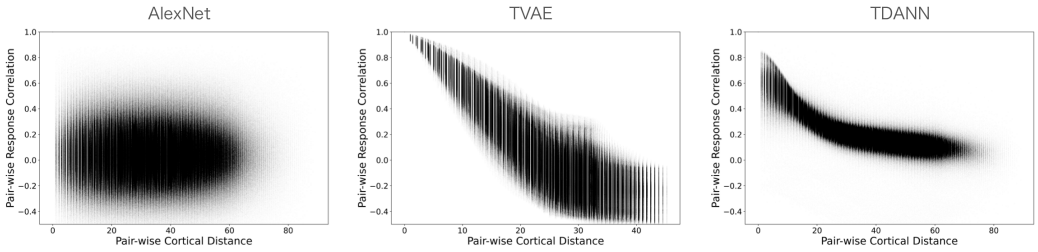


Figure A.3: Pairwise correlation between neurons as a function of distance in the cortical sheet.

*Impact of TVAE Spatial Window Size (**W**)*

In Figure A.4 below, we demonstrate the effect of different choices of topographic organization (defined by **W**) on the resulting learned selectivity maps. Specifically, we keep the global topography the same (a 2-d grid with circular boundary conditions), but we change the spatial extent over which variance is shared between variables **t**. From left to right, we defined the matrix **W** to be a convolution matrix with kernels of size 5×5, 15×15, 25×25, and 35×35, where the total grid size is 64×64.

TDANN Nested Spatial Hierarchy

In Figure A.5 below, we show the abstract selectivity maps for the TDANN, analogous to those in Figure 4.5 for the TVAE in the main paper. We see that the TDANN does appear to have a similar nested spatial hierarchy, however it is difficult to mea-

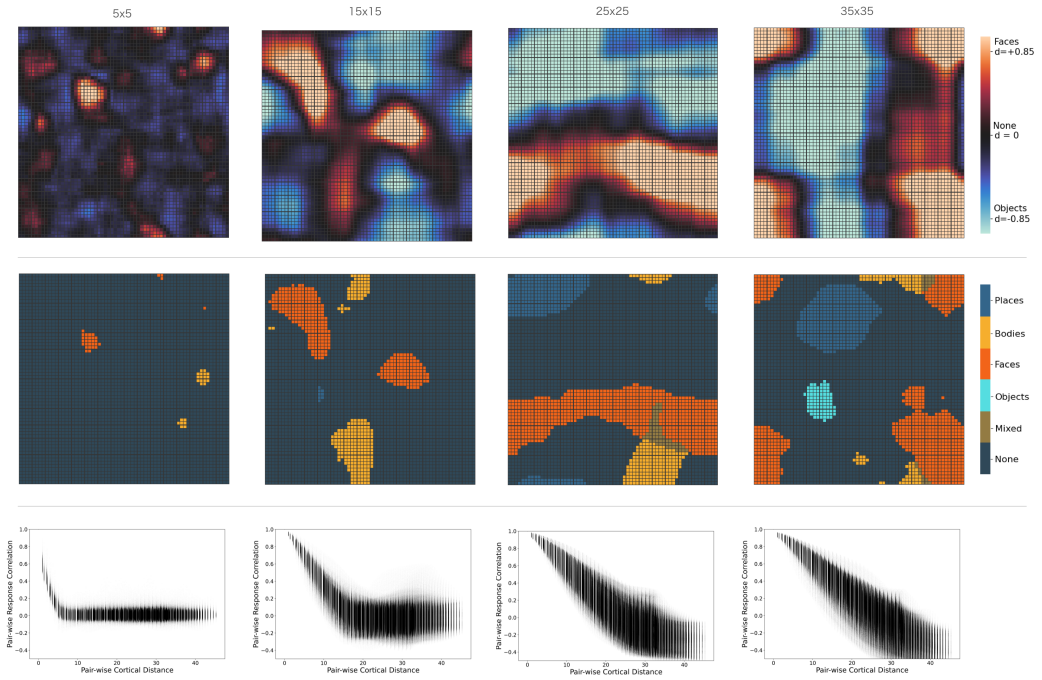


Figure A.4: Selectivity maps and pairwise correlation curves for different choices of spatial window size in the Topographic VAE.

sure the differences visually. In future work, we hope to explore methods for quantifying the coherence of selectivity hierarchies, allowing greater comparison of models on this front.

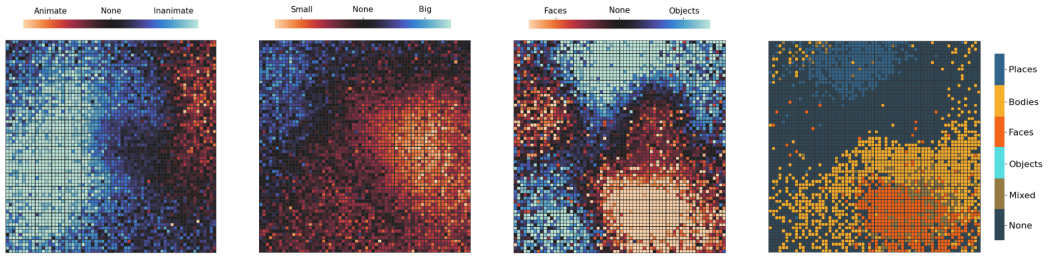


Figure A.5: Abstract category selectivity for the TDANN, analogous to the results presented in Figure 4.5 for the TVAE. From left to right: Animate vs. Inanimate, Small vs. Big, Faces vs. Objects, and Multi-class selectivity with $d \geq 0.85$ (analagous to Figure 4.3).

VAE Baseline

As an additional non-topographic baseline, we train a standard VAE in-place of the TVAE and measure the selectivity and single-image activation maps as in Figures 4.2

and 4.4. Interestingly, we see that the standard VAE exhibits significantly fewer class-selective neurons, with the majority of neurons activating for each image. We find this correlates with the measured likelihood of the data under each model, suggesting that topographic organization (and similarly class-selectivity) acts as regularization on model performance, slightly reducing the overall likelihood. As measured in prior work **leavitt2020selectivity**, high class-selectivity is similarly seen to be slightly detrimental to classification performance, agreeing with these results.

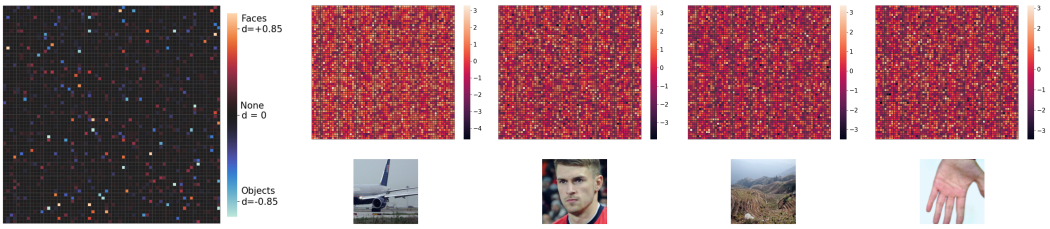


Figure A.6: Face vs. Object selecitivity (left) and single-image activation maps (right) for a non-topographic VAE baseline

Appendix B

CHAPTER V APPENDIX

B.1. Experiment Details

Code for reproducing the original Topographic VAE experiments can be found at: `https://github.com/AKAndyKeller/TopographicVAE`.

Code for reproducing the Predictive Coding Topographic VAE experiments can be found at: `https://github.com/akandykeller/PCTVAE`.

Optimizer Parameters

All models were trained with stochastic gradient descent on batches of size 8 (due to each batch-example being a length 15 or 18 sequence), using a learning rate of 1×10^{-4}, and standard momentum of 0.9 for 100 epochs.

Initalization

All weights of the models were initialized with uniformly random samples from $U(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$, where m is the number of input units. For all topographic models including BubbleVAE, μ was initialized to a large value (30.0) as this was observed to increase the speed of convergence and was sometimes necessary for observed topographic organization in deeper models.

Model Architectures

All models presented in this paper make use of the same 3-Layer MLP for parameterizing the encoders and decoders. Specifically, the model is constructed as 3 fully connected layers with ReLU activations in-between the layers. For MNIST, the layers of both the $\mathbf{u}$ and $\mathbf{z}$ encoders have (972, 648, 648) output units each for the first, second, and third layers respectively. The 648 units in the third layer are divided into two sets to compute the mean and log standard deviation of the respective u's and z's, yielding 324 t variables. This is then divided into 18 capsules, each of 18 dimensions. The layers of the decoder have (648, 972, 2352) output units respectively. For dSprites, both encoder layers have output sizes (674, 450, 450), where the re-

sulting 225 t variables are divided into 15 capsules, each of 15 dimensions. The decoder layers then have output sizes (450, 675, 4096). We note the non-topographic VAE baselines make use of only a single encoder for the Gaussian variable $\mathbf{z}$ (as $\mathbf{u}$ is not needed), and do not incorporate a μ parameter.

Choice of $\mathbf{W}$

For the TVAE and BubbleVAE, the global topographic organization afforded by $\mathbf{W}$ was fixed to a set of 1-D tori ('circular capsules') as depicted in Figure 5.2. Practically, multiplication by $\mathbf{W}$ was performed by convolution over the appropriate dimensions (time & capsule dimension) with a kernel of all 1's, taking advantage of circular padding to achieve toroidal structure.

Choice of $\mathbf{W}_\delta$

The choice of $\mathbf{W}_\delta$ determines the local topographic structure within a single timestep. For all TVAE models with $L > 0$, we experimented with local neighborhood sizes (denoted K) of 3 units (effective kernel size 3 in the capsule dimension), and 1 unit (no neighborhood). For MNIST it was observed that $K = 3$ performed best, while $K = 1$ worked best for dSprites. This is likely due to the slower, smoother, and more overlapping transformations constructed on MNIST, whereas our subset of dSprites contained non-smooth transformations where the overlap between successive images was smaller (e.g. due to sub-setting, see Section B.1), which made larger neighborhood sizes $K > 1$ less fitting. For TVAE models with $L = 0$, $\mathbf{W}_\delta = \mathbf{W}$ was fixed to sum over neighborhoods of size $K = 9$ for MNIST and $K = 3$ for dSprites. These values were chosen to be sufficiently large to achieve notably lower equivariance error than the VAE baseline, and thus demonstrate the impact of topographic organization without temporal coherence. For BubbleVAE models, the extent of topographic organization in the capsule dimension was set to $K = 3$ on MNIST to match the TVAE, and was set to be equal to the organization in time dimension $K = 2L$ for dSprites. A further quantitative comparison on the impact of the choice of the K parameter can be found in Section B.2.

Choice of L

The choice of L determines the extent of temporal coherence where $2L$ equals the input sequence length, and $L = 0$ corresponds to single inputs. For Table 1,

we experimented with values of L in the set $\{0, \frac{5}{36}S, \frac{1}{4}S, \frac{1}{2}S\}$ for both the TVAE and BubbleVAE. Both the BubbleVAE and TVAE achieved highest likelihoods at $L = \frac{5}{36}S$, and TVAE achieved lowest equivariance error at $L = \frac{1}{2}S$. We additionally included TVAE experiments with $L = \frac{13}{36}S$ for purposes of visualization in Figures 1 and 4 as this yielded the best qualitative generalization. For Table 2, we experimented with values of L in the set $\{0, \frac{1}{6}S, \frac{4}{15}S, \frac{1}{3}S, \frac{2}{5}S, \frac{1}{2}S\}$ for both TVAE and BubbleVAE, and presented a broad selection in the table. The results of all models are shown in Section B.2 below.

Hyperparameter Selection

Hyperparameters such as learning rate, batch size, number of capsules, capsule size, and ultimately model architecture were chosen to allow for quick training on limited resources and were not tuned significantly. Since it was conceptually simpler to have an equal number of capsule dimensions and sequence elements, this limited the number of capsules we could then train efficiently. In Section B.3 we explain how a model with fewer capsule dimensions than sequence elements could be constructed with an alternative Roll operator. Additionally, from preliminary experiments, we observe that models with a number of internal capsule dimensions different from the number of sequence elements achieve similar likelihood values while also learning coherent transformations as decoded through the capsule roll. We believe these findings in combination with the extra studies provided in Section B.2 suggest a satisfying degree of robustness to hyperparameter selection.

MNIST Transformations

The first set of experiments presented in this paper are based on the MNIST dataset (LeCun, Cortes, and Burges, 2010) (MIT Licence). For Section 6.2 (Figure 3) an MNIST training set of 48,000 images was used, while the standard test set of 10,000 images was used to compute the maximum activating image. For Section 6.3 (Figure 4 and Table 1), sequences of MNIST images were created by picking a random training image (with a random transformation 'pose') and successively transforming it according to one of the 3 available transformations (e.g. only one attribute is changed per sequence). The available transformations consisted of rotation, color (hue rotation), and scale with increments of 20-degrees for rotation and color, and 3.66% increments for scale. Since scale is inherently non-cyclic, the bounds of the transformation were set at 60% and 126%, and the transformations were constructed

to be periodic such then once scale reached 126%, the next element was at 60% scale. The final sequences were thus constructed to be 18 images long, where each element in the batch had an independently randomly chosen transformation. Again, the likelihood $\log p(\mathbf{x})$ and equivariance error $\mathcal{E}_{eq}$ were computed on the held-out 10,000 example test set, where the same random transformation sequences were applied.

dSprites Transformations

The second set of experiments presented in this paper are based on the dSprites dataset (Matthey et al., 2017) (Apache-2.0 License). To reduce computational complexity of this dataset, we took a subset of the dataset which consisted of all 3 shapes, the largest 5 scales, and every other example from the first 30 orientations, x-positions, and y-positions. The resulting dataset thus had 50,625 total images (3 shapes, 5 scales, 15 orientations, 15 x-positions, 15 y-positions), compared to the original 737,280 images. To construct sequences, we followed the same procedure as for MNIST, whereby first a random example and transformation were chosen, and a sequence of 15 images was constructed where only the chosen transformation was applied successively. We define the transformations available for sequences as scale, orientation, x-position, and y-position, omitting shape since smooth shape transforms are not present in the dSprites dataset. Again, we define all transformations to be cyclic such that once the 15th element is reached, the 1st element follows. For scale transformations, we simply loop over all 5 scales 3 times per sequence. We observe that although these sequences do not match the latent priors exactly, the models still train relatively well, implying some degree of robustness.

*Capsule Correlation Metric (*CapCorr*)*

Here we define CapCorr more precisely as it is implemented in our work. First, we denote the ground truth transformation parameter of the sequence at timestep l as y_l (e.g. the rotation angle at timestep l for a rotation sequence), and the corresponding activation at time l as $\mathbf{t}_l$. Next, to get an arbitrary starting point, we let $l = \Omega$ denote the timestep when y_l is at its canonical position (e.g. rotation angle 0, x-position 0, or scale 1). We see Ω is not necessarily 0 since the first timestep of each sequence ($l = 0$) is a randomly transformed example. Then, we observe that we can measure the approximate observed roll in the capsule dimension between time 0 and Ω as a 'phase shift' by computing the index of the maximum value of a discrete (periodic)

cross-correlation of $\mathbf{t}_\Omega$ and $\mathbf{t}_0$:

$$\text{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0) = \text{argmax}\left[\mathbf{t}_\Omega \star \mathbf{t}_0\right] \tag{B.1}$$

Where $\star$ is discrete (periodic) cross-correlation across the (cyclic) capsule dimension and argmax is also subsequently performed over the capsule dimension. Then, the CapCorr metric for a single capsule is given as:

$$\text{CapCorr}(\mathbf{t}_\Omega, \mathbf{t}_0, y_\Omega, y_0) = \text{Corr}\left(\text{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0), |y_\Omega - y_0|\right) \tag{B.2}$$

Where the correlation coefficient Corr is then computed across all examples for the entire dataset. In our experiments we use the Pearson correlation coefficient for Corr. We thus see this metric is the correlation of the estimated observed capsule roll with the shift in ground truth generative factors, which is equal to 1 when the model is perfectly equivariant. To extend this definition to multiple capsules, we estimate ObservedRoll for each capsule separately, and then correlate the mode of all ObservedRoll values with the true shift in ground truth generative factors. We see empirically that the ObservedRolls for all capsules are almost always identical (i.e. all capsules roll simultaneously for each transformation), therefore computing the mode does not destroy significant information. Finally, for transformation sequences which have multiple timesteps where y_l is at the canonical position (e.g. scale transformations on dSprites where scale is looped 3 times), we select $l = \Omega$ to be the one from this possible set which yields the minimal absolute distance between $|y_\Omega - y_0|$ and $\text{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0)$.

Definition of Roll *for Capsules*

As stated in Section 4.5.2, $\text{Roll}_\delta(\mathbf{u})$, is defined as a cyclic permutation of δ steps along the capsule dimension of $\mathbf{u}$. Explicitly, if $\mathbf{u}$ is divided into C capsules each with D dimensions, the Roll_δ operation can be written as:

$$\begin{aligned}\text{Roll}_\delta(\mathbf{u}) &= \text{Roll}_\delta\left([u_1, u_2, \ldots, u_{C\cdot D}]\right) \\ &= [u_D, u_1, \ldots, u_{D-1}, u_{2\cdot D}, u_{D+1}, \ldots, u_{2\cdot D-1}, u_{3\cdot D}, \ \ldots \ , \ \ldots \ u_{C\cdot D-1}]\end{aligned} \tag{B.3}$$

B.2. Extended Results

In this section we provide extended results for all tested hyperparamters (Tables B.1 & B.2), a further analysis of the impact of the coherence window within a capsule

Table B.1: Log Likelihood and Equivariance Error on MNIST for all models tested. Mean ± std. over 3 random initalizations.

Model	TVAE	TVAE	TVAE	TVAE	TVAE
L	$L = \frac{1}{2}S$	$L = \frac{13}{36}S$	$L = \frac{1}{4}S$	$L = \frac{5}{36}S$	$L = 0$
K	$K = 3$	$K = 3$	$K = 3$	$K = 3$	$K = 9$
$\log p(\mathbf{x}) \uparrow$	$\mathbf{-186.8} \pm 0.1$	-188.0 ± 0.5	-187.0 ± 0.2	$\mathbf{-186.0} \pm 0.7$	-218.5 ± 0.9
$\mathcal{E}_{eq} \downarrow$	$\mathbf{573.9} \pm 1.5$	1089.8 ± 2.4	2136.9 ± 7.8	3246.6 ± 3.3	3216.6 ± 104.9

Model	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	VAE
L	$L = \frac{1}{2}S$	$L = \frac{1}{4}S$	$L = \frac{5}{36}S$	$L = \frac{5}{36}S$	$L = 0$
K	$K = 2L$	$K = 2L$	$K = 2L$	$K = 3$	$K = 1$
$\log p(\mathbf{x}) \uparrow$	-200.9 ± 0.7	-202.3 ± 1.4	-190.8 ± 0.7	-191.4 ± 0.5	-189.0 ± 0.8
$\mathcal{E}_{eq} \downarrow$	4206.7 ± 903.3	1141.7 ± 9.6	2605.7 ± 16.1	3369.5 ± 11.9	13273.9 ± 0.5

$\mathbf{W}_\delta$ (Table B.3), samples from the model in Section 4.5, and additional capsule traversal experiments highlighting the generalization capabilities of the TVAE to combinations of transformations unseen during training (Figure B.2).

Extended Tables 1 & 2

In Tables B.1 & B.2 below, we present extended versions of Tables 1 & 2 respectively, showing all tested settings of the TVAE & BubbleVAE. We observe the TVAE achieves perfect correlation (CapCorr = 1) for $L \geq \frac{1}{3}$, and steadily decreasing correlation for lower values of L.

Impact of $\mathbf{W}_\delta$

In Table B.3, we show a small set of experiments with different settings of $\mathbf{W}_\delta$, and specifically changing values of K (the coherence window within a capsule). As can be seen, increasing K generally reduces equivariance error, but decreases the log-likelihood. This can be further understood by examining the capsule traversals of such models in Figures B.5, B.6, B.7, B.8, & B.9. We see that larger values of K appear to induce smoother transformations within the capsule dimensions, eventually resulting in invariant representations when K is equal to the capsule dimensionality.

Table B.2: Equivariance error and CapCorr for all models tested on the dSprites dataset. Mean ± standard deviation over 3 random initalizations.

Model	TVAE	TVAE	TVAE	TVAE	TVAE	TVAE
L	$L = \frac{1}{2}S$	$L = \frac{2}{5}S$	$L = \frac{1}{3}S$	$L = \frac{4}{15}S$	$L = \frac{1}{6}S$	$L = 0$
K	$K = 1$	$K = 1$	$K = 1$	$K = 1$	$K = 1$	$K = 3$
$\text{CapCorr}_X \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.95 ± 0.00	0.67 ± 0.02	0.17 ± 0.03
$\text{CapCorr}_Y \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.96 ± 0.01	0.66 ± 0.02	0.21 ± 0.02
$\text{CapCorr}_O \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.88 ± 0.01	0.52 ± 0.01	0.09 ± 0.01
$\text{CapCorr}_S \uparrow$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	$\mathbf{1.0} \pm 0$	0.96 ± 0.01	0.42 ± 0.01	0.51 ± 0.01
$\mathcal{E}_{eq} \downarrow$	$\mathbf{344} \pm 5$	759 ± 9	1034 ± 6	1395 ± 7	2549 ± 38	2971 ± 9

Model	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	BubbleVAE	VAE
L	$L = \frac{1}{2}S$	$L = \frac{2}{5}S$	$L = \frac{1}{3}S$	$L = \frac{4}{15}S$	$L = \frac{1}{6}S$	$L = 0$
K	$K = 2L$	$K = 2L$	$K = 2L$	$K = 2L$	$K = 2L$	$K = 1$
$\text{CapCorr}_X \uparrow$	0.16 ± 0.01	0.15 ± 0.01	0.13 ± 0.01	0.12 ± 0.02	0.09 ± 0.01	0.18 ± 0.01
$\text{CapCorr}_Y \uparrow$	0.15 ± 0.01	0.14 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.11 ± 0.02	0.16 ± 0.01
$\text{CapCorr}_O \uparrow$	0.12 ± 0.00	0.13 ± 0.02	0.10 ± 0.01	0.09 ± 0.00	0.06 ± 0.01	0.11 ± 0.00
$\text{CapCorr}_S \uparrow$	0.52 ± 0.02	0.55 ± 0.00	0.52 ± 0.00	0.48 ± 0.02	0.27 ± 0.01	0.52 ± 0.00
$\mathcal{E}_{eq} \downarrow$	6825 ± 126	6917 ± 13	1951 ± 34	2181 ± 627	1721 ± 27	6934 ± 0

Table B.3: Impact of $\mathbf{W}_\delta$ (i.e. K) on MNIST performance.

Model	TVAE	TVAE	TVAE	TVAE	TVAE
L	$L = \frac{5}{36}S$	$L = \frac{5}{36}S$	$L = 0$	$L = 0$	$L = 0$
K	$K = 3$	$K = 9$	$K = 3$	$K = 9$	$K = 18$
$\log p(\mathbf{x}) \uparrow$	$\mathbf{-186.0} \pm 0.7$	-190.6 ± 0.2	-213.4 ± 1.2	-218.5 ± 0.9	-224.8 ± 1.0
$\mathcal{E}_{eq} \downarrow$	3246.6 ± 3.3	2606.3 ± 17.0	12085.7 ± 68.5	3216.6 ± 104.9	1090.3 ± 19.3

Samples

In Figure B.1, we provide samples from our model in the $L = 0$ setting to validate that the learned latent distribution closely matches the $TPoT$ distribution described in Equation 4.6. Explicitly, the samples are generated by sampling standard normal random variables $\mathbf{Z}$ and $\mathbf{U}$, constructing $\mathbf{T}$ as in Equation 4.6, and then passing these sampled $\mathbf{T}$ through the decoder. We see that the samples resemble true MNIST digits (accounting for the limited capacity of the model), implying that the distribution after training indeed follows the desired distribution, and the model has learned to become a good generative model of the data.

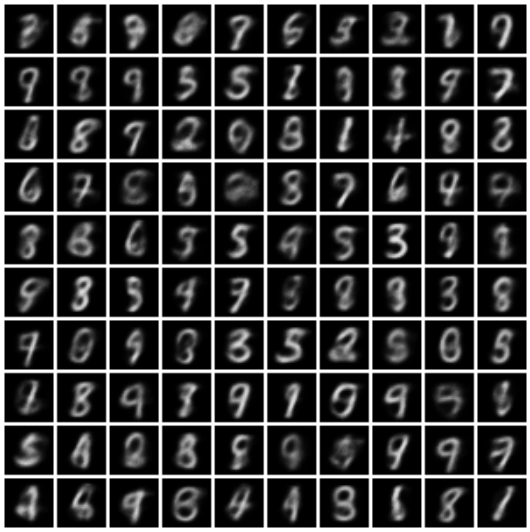


Figure B.1: Samples from the TVAE in Section 4.5.

Generalization to Combined Transformations at Test Time

In this section, we test the ability of the model to generate sequences composed of multiple transformations through a capsule roll, despite only being trained on individual transformations in isolation. In other words, we intend to measure the extent to which the transformations learned by a set of capsules can be combined simply by passing input sequences with corresponding combined transformations. Such generalization suggests powerful benefits to data efficiency, effectively factorizing a set of complex transformations.

Explicitly, we train the model identically to that presented in Figure 5.3, (TVAE $L = \frac{13}{36}S$), and examine the sequences generated by a capsule roll when the partial input sequences contain combinations of transformations previously unseen during training. The results of this experiment, tested on combinations of rotation and color transforms on the MNIST test set, are presented in Figure B.2 below. Although this generalization capability is not known to be guaranteed a priori, we see that the capsule traversals are frequently remarkably coherent with the input transformation, implying that the model may indeed be able to generalize to combinations of transformations. Furthermore, we observe with $L = \frac{1}{2}S$ (results not shown), this generalization capability is nearly perfect.

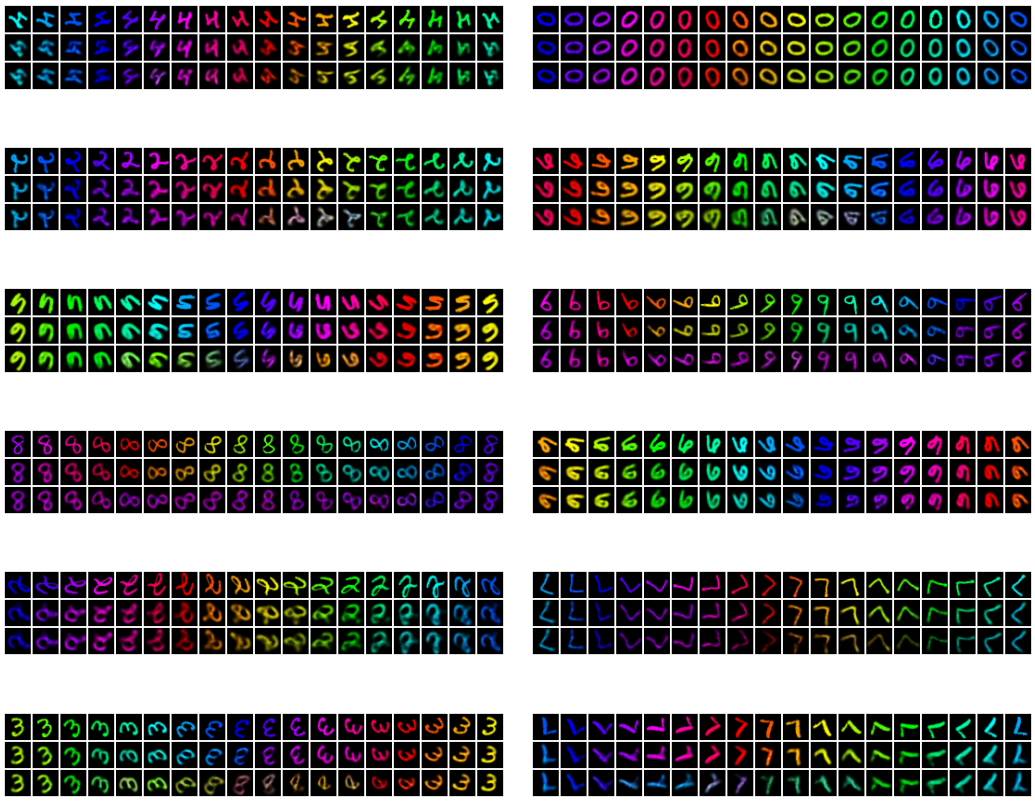


Figure B.2: Capsule Traversals for MNIST TVAE $L = \frac{13}{36}S$, trained on individual transformations in isolation, and tested on combined color and rotation transformations. Top row shows the input sequence, middle row shows the direct reconstruction $\{g_\theta(\mathbf{t}_l)\}_l$, and bottom row shows the capsule traversal $\{g_\theta(\text{Roll}_l[\mathbf{t}_0])\}_l$.

B.3. Proposed Model Extensions

Extensions to Roll *&* CapCorr

The Roll operation can be seen as defining the speed at which $\mathbf{t}$ transforms corresponding to an observed transformation. For example, with Roll defined as in Section B.1 above, we implicitly assume that for each observed timestep, we would like the representation $\mathbf{t}$ to cyclically permute 1-unit within the capsule. For this to match the observed data, it requires the model to have an equal number of capsule dimensions and sequence elements. If we wish to reduce the size of our representation, we could instead encourage a 'partial permutation' for each observed transformation. For a single capsule with D elements, an example of a simple linear version of such a partial permutation (for $0 < \alpha \leq 1$) can be implemented as:

$$\text{Roll}_\alpha(\mathbf{u}) = \left[\alpha u_D + (1-\alpha)u_1,\ \ \alpha u_1 + (1-\alpha)u_2,\ \ \ldots,\ \ \alpha u_{D-1} + (1-\alpha)u_D\right] \quad \text{(B.4)}$$

A slightly more principled partial roll for periodic signals could also be achieved by performing a phase shift of the signal in Fourier space, and performing the inverse Fourier transform to obtain the resulting rolled signal. To extend the CapCorr metric to similarly allow for partial Rolls, we see that we can simply redefine the ObservedRoll (originally given by discrete cross-correlation) to be given by the argmax of the inner product of a sequentially partially rolled activation with the initial activation $\mathbf{t}_\Omega$. Formally:

$$\text{ObservedRoll}(\mathbf{t}_\Omega, \mathbf{t}_0) = \text{argmax}\left[\mathbf{t}_\Omega \cdot \text{Roll}_0(\mathbf{t}_0),\ \mathbf{t}_\Omega \cdot \text{Roll}_\alpha(\mathbf{t}_0),\ \ldots\ , \mathbf{t}_\Omega \cdot \text{Roll}_{D-\alpha}(\mathbf{t}_0)\right] \tag{B.5}$$

Non-Cyclic Capsules

We can also see that there is nothing beyond convenience which inherently requires the capsules to be circular (i.e. have periodic boundary conditions). To implement linear capsules, we propose one solution is to add L additional U_i variables to both the left and right boundaries of each capsule. In this way, the vector $\mathbf{U}$ is larger than the vector $\mathbf{Z}$ and can be seen as a 'padded' version, where the padding is composed of independant random variables. Additionally, the transformation sequences can then be padded on both sides by replicating the first and final elements L times. The construction of $\mathbf{T}$ variables is then performed identically as in Equations 5.9 and 5.10. The Roll operation can then be similarly defined as filling the boundaries with 0 since these values will not be used as part of the computation.

Multi-dimensional Temporally Coherent Capsules

In consideration of transformations which may naturally live in multiple dimensions, we wish to extend the original model to support multi-dimensional capsules. Such multi-dimensional capsules could additionally support more well-defined 'disentanglement' of transformations by encouraging each transformation to be axis-aligned with one dimension of each capsule. We see that in the non-temporally coherent case ($L = 0$), the model can easily be extended to capsules of multiple dimensions through multi-dimensional neighborhoods. An example of a model with 2-dimensional neighborhoods is presented in Figure 3. However, when considering shifting temporal coherence as we defined in Section 6.3, it is not clear how the shift operator or the neighborhoods should be defined for higher dimensional capsules. In this section we propose to modify the definitions of $\mathbf{T}$ in Equations 5.9 and 5.10 with an extension resembling 'group sparsity' in the denominator.

First, we again assume that each input sequence is an observation of a single transformation at a time. Formally, the multi-dimensional capsules are then constructed by arranging $\mathbf{U}$ into a D dimensional lattice. In such a model, we desire to roll and sum only along a single axis of the lattice for a given sequence. Incorporating this into the construction of $\mathbf{T}$ yields the following:

$$\mathbf{T}_l = \frac{\mathbf{Z}_l - \mu}{\sum_{d=1}^{D} \sqrt{\mathbf{W}^d \left[\mathbf{U}^2_{l+L}; \cdots; \mathbf{U}^2_{l-L}\right]}} = \frac{\mathbf{Z}_l - \mu}{\sum_{d=1}^{D} \sqrt{\sum_{\delta=-L}^{L} \mathbf{W}^d_\delta \text{Roll}^d_\delta(\mathbf{U}^2_{l+\delta})}} \tag{B.6}$$

Where $\mathbf{W}^d_\delta$ refers to a matrix which sums locally along the d^{th} dimension of each capsule, and not at all along the others, and similarly Roll^d_δ rolls only along the d^{th} dimension. In practice we observe such models can indeed disentangle up to 2 distinct transformations, but become more challenging to optimize for higher dimensions. We believe this is potentially due to the exponential growth in capsule size with increasing dimension, but leave further exploration to future work.

B.4. Capsule Traversals

In this section we provide a set of 12 capsule traversals for each of the models presented in main text. The traversals are randomly selected such that all transformations (and dSprites shapes) are shown evenly. Unlike the main section, we additionally include a middle row which shows the direct reconstruction of the input without any rolling (i.e. $\{g_\theta(\mathbf{t}_l)\}_l$). We find the direct reconstructions valuable to determine if poor traversals are due to bad reconstructions (low $\log p_\theta(\mathbf{x}|\mathbf{t})$) or a lack of equivariance (high $\mathcal{E}_{eq}$). For example, with the baseline VAE models, we see that the reconstructions in the middle row are accurate for the full sequence, while the capsule traversals obtained by sequentially rolling the initial activation (shown in the bottom row) are nothing like the input transformation (top row). In all traversals, the left-most image corresponds to $\mathbf{t}_0$, and thus input sequences of length $2L$ cover both the left and right edges when $L > 0$.

Finally, in Figures B.18 & B.19 at the end of the section, we include capsule traversals for models trained on MNIST with more complex transformations such as combined color & rotation, and combined color & perspective transforms. These models were trained in an identical manner to the other MNIST models, with the same architecture, only changing the transformation sequences of the training dataset.

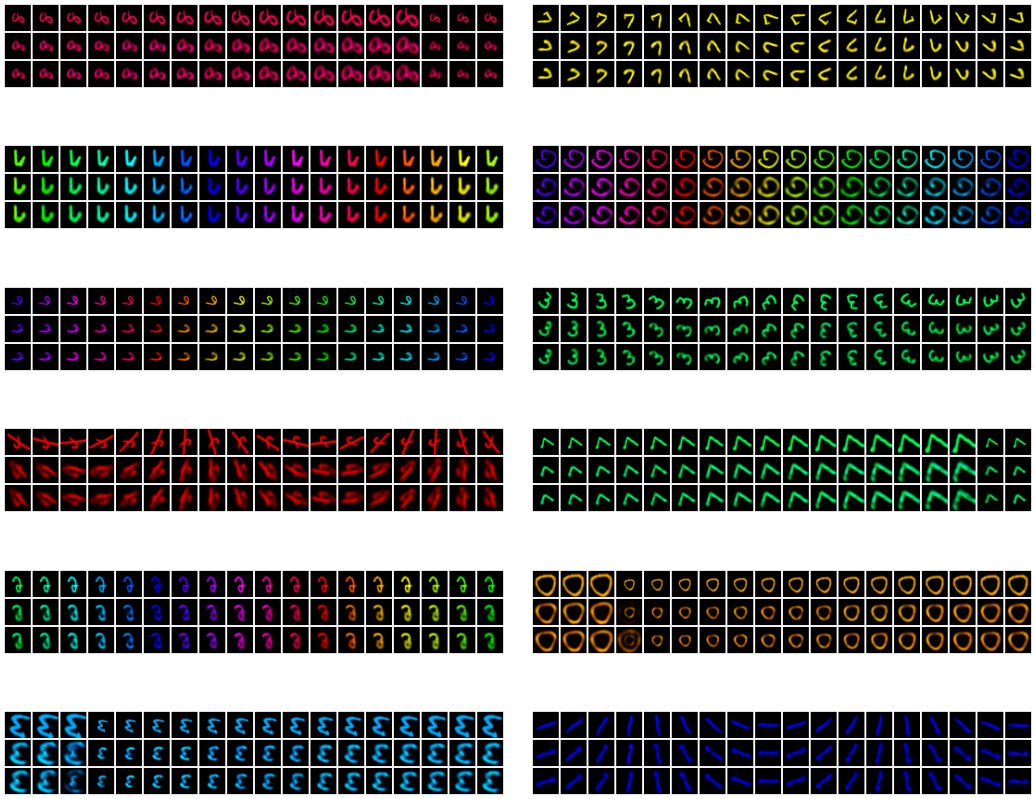


Figure B.3: MNIST TVAE $L = \frac{1}{2}S$, $K = 3$

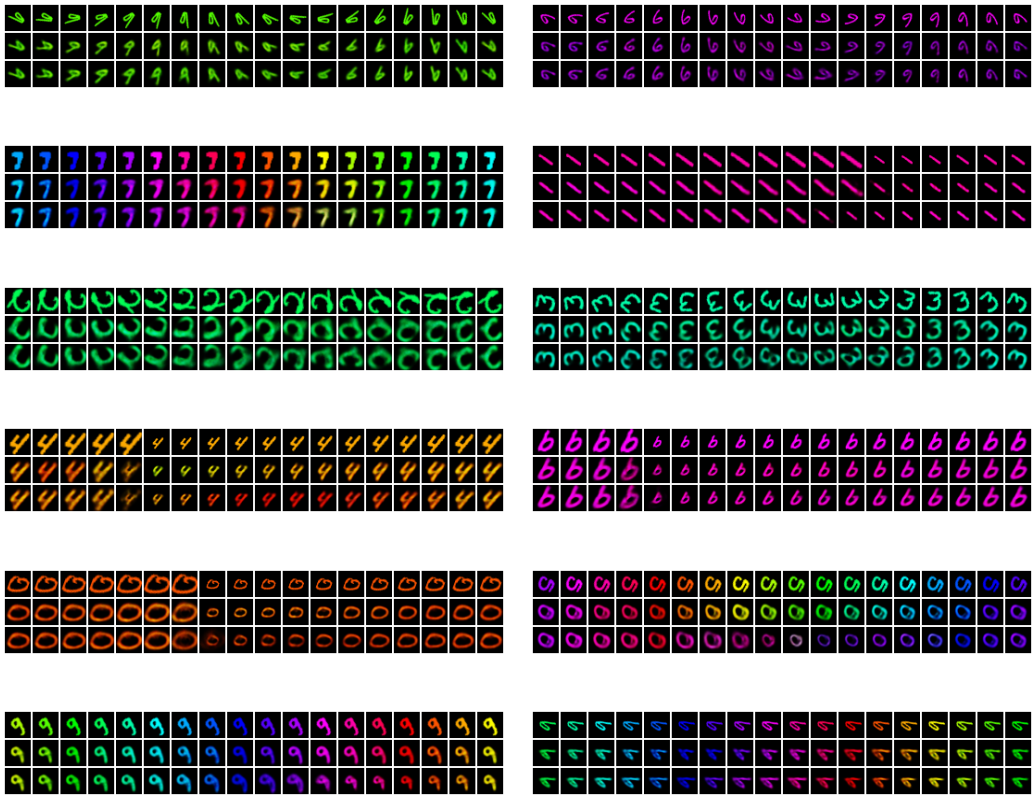


Figure B.4: MNIST TVAE $L = \frac{13}{36}S$, $K = 3$

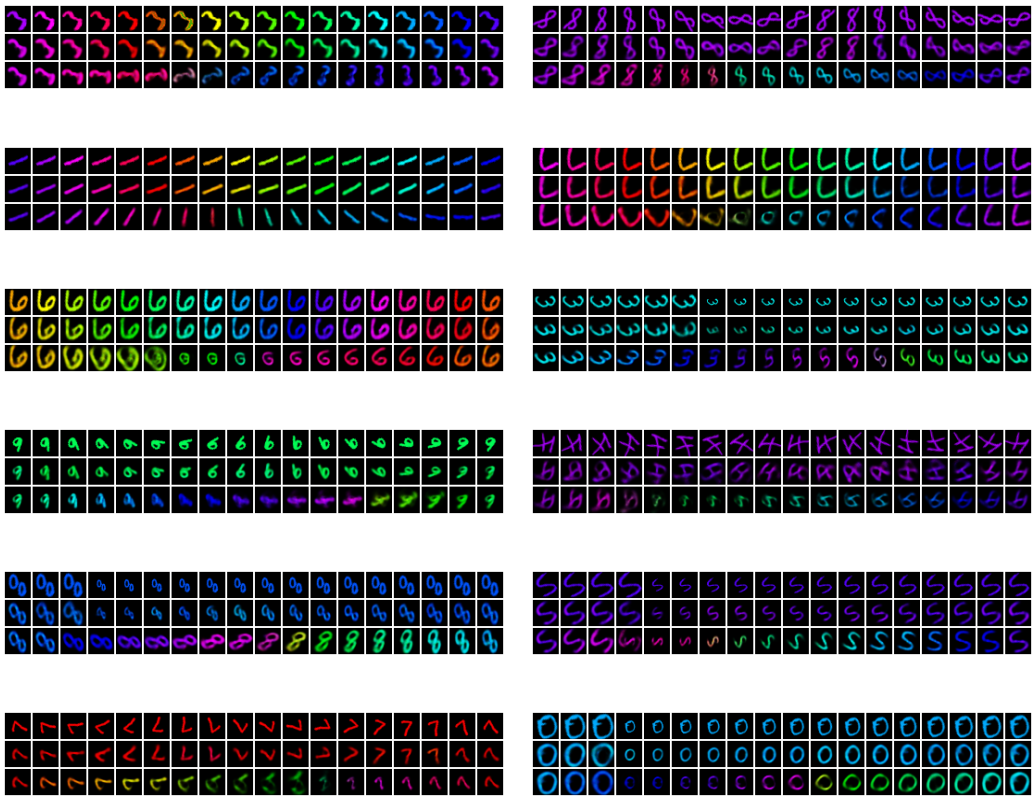


Figure B.5: MNIST TVAE $L = \frac{5}{36}S$, $K = 3$. We see with values of $L < \frac{1}{3}S$ the transformations decoded through the capsule roll are only partially coherent with the input sequence.

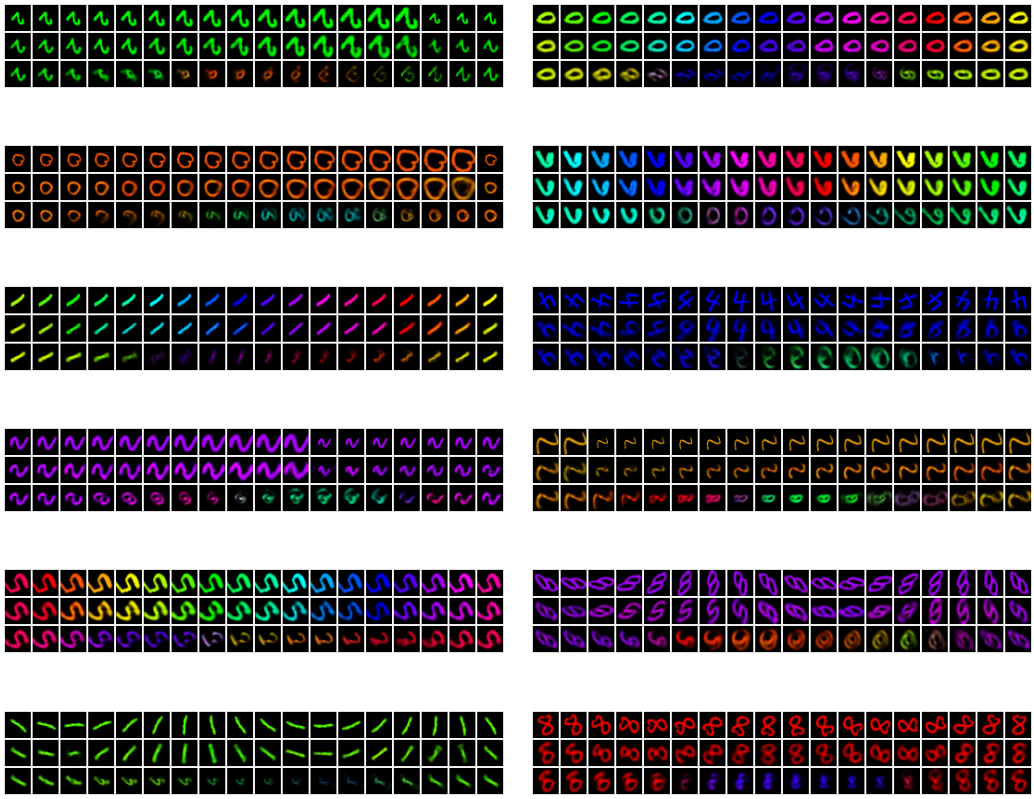


Figure B.6: MNIST TVAE $L = \frac{5}{36}S$, $K = 9$

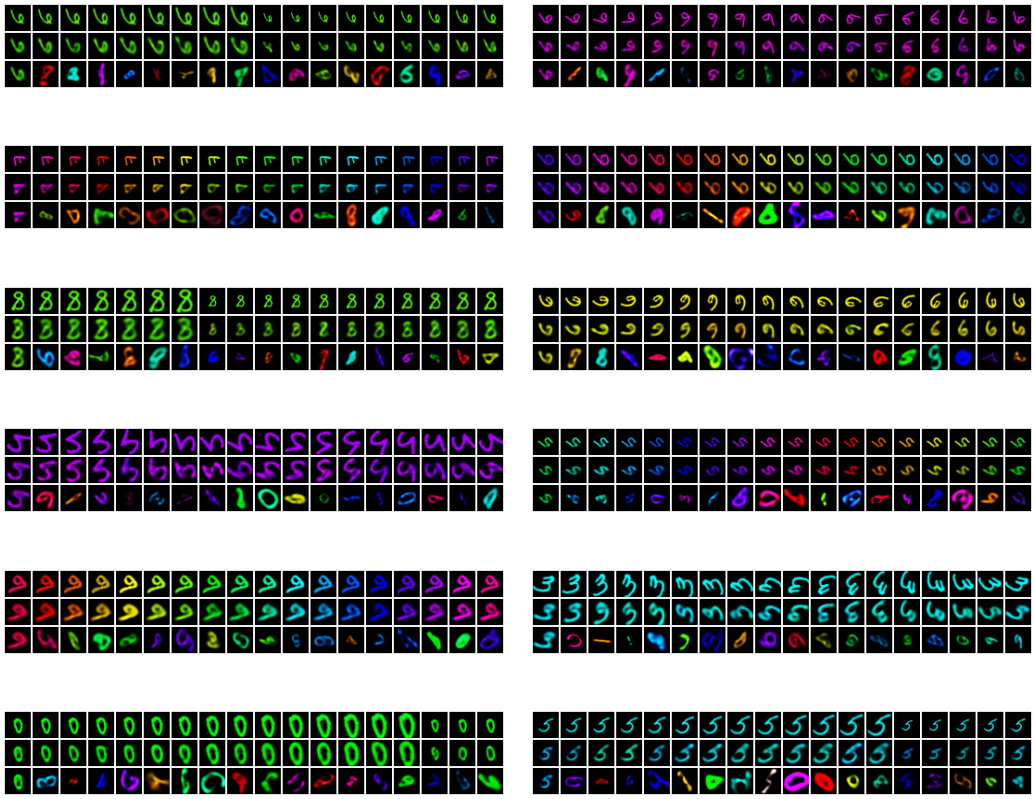


Figure B.7: MNIST TVAE $L = 0$, $K = 3$. We see for sufficiently small values of K, the TVAE can reach a degenerate solution where topographic organization is almost entirely lost.

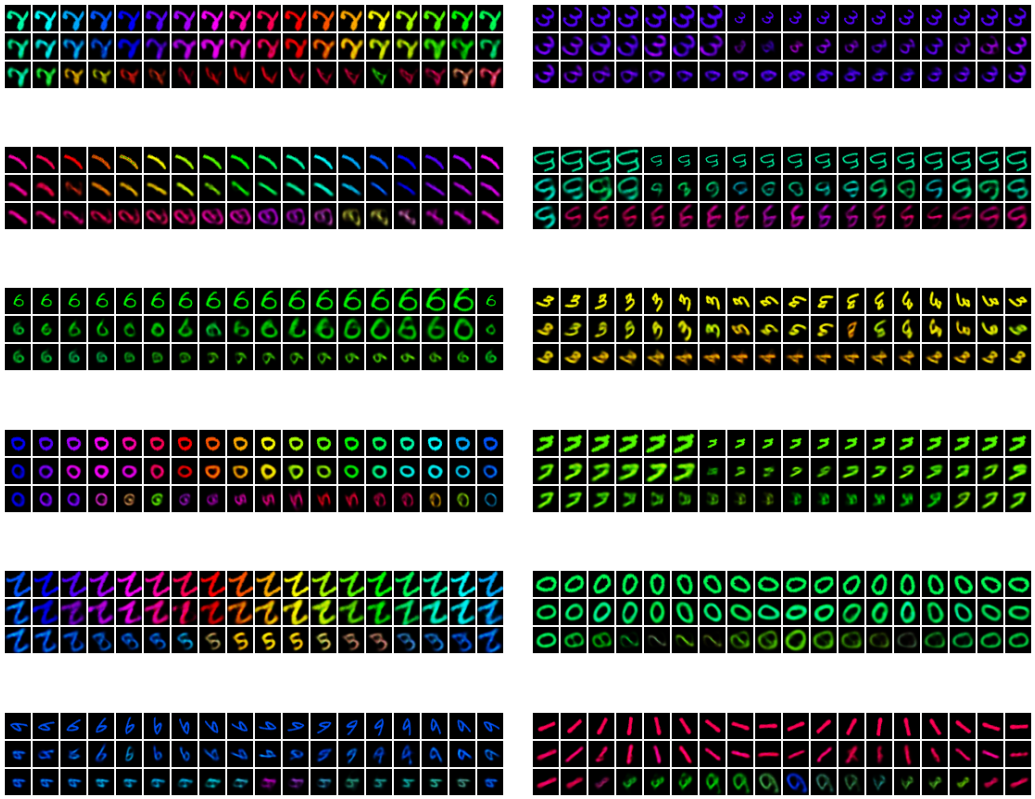


Figure B.8: MNIST TVAE $L = 0$, $K = 9$

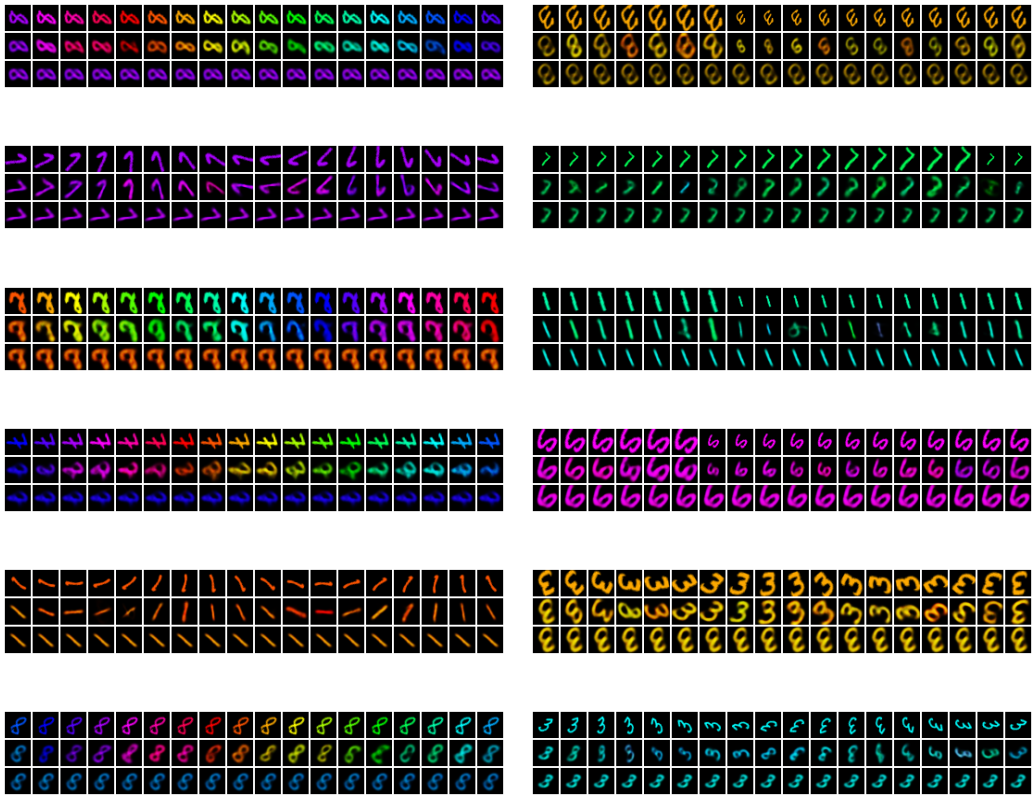


Figure B.9: MNIST TVAE $L = 0$, $K = 18$. We see when K is equal to the capsule size (making the model analogous to ISA), the model learns an invariant capsule representation – meaning Rolling a capsule activation produces no significant transformation in the observation space.

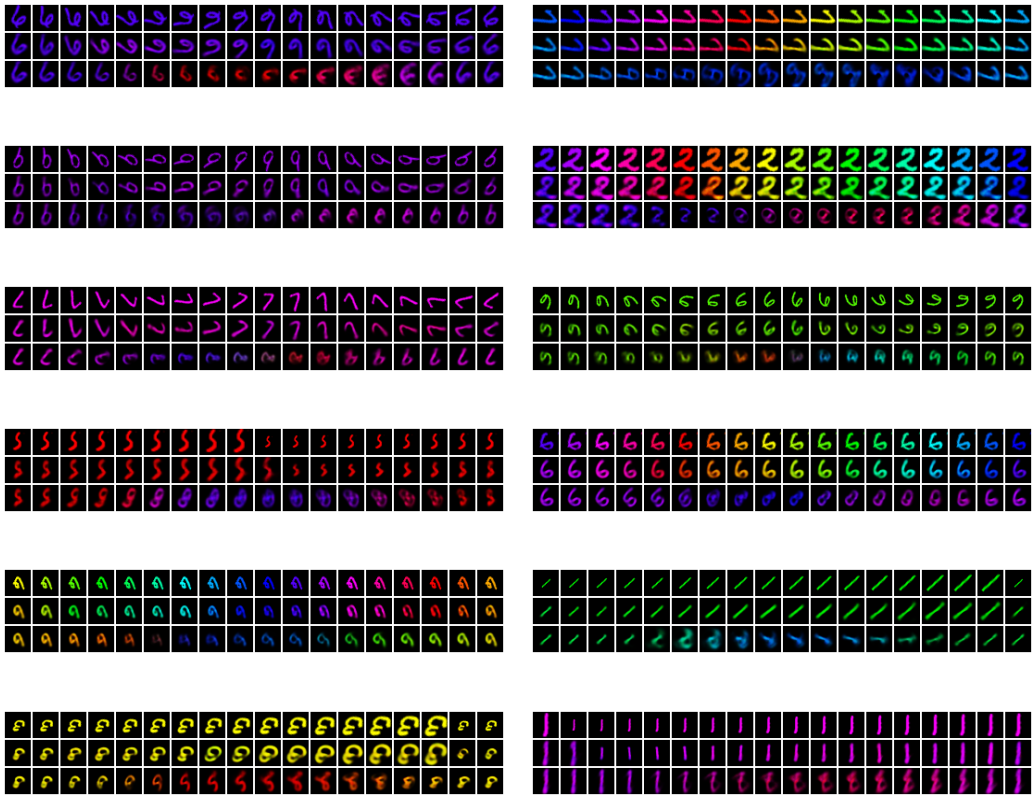


Figure B.10: MNIST BubbleVAE $L = \frac{5}{36}S$, $K = 2L$

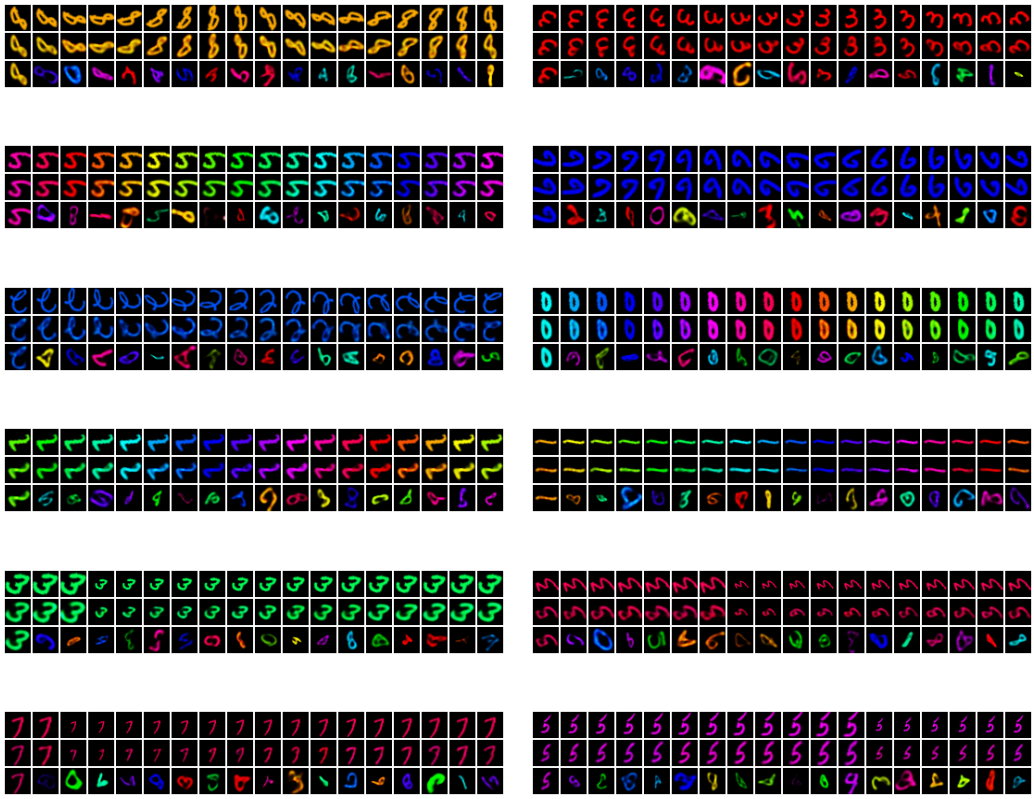


Figure B.11: MNIST VAE $L = 0$, $K = 1$. We see images generated through capsule traversal with the baseline VAE appear entirely random, as expected due to the non-topographic nature of the VAE's latent space.

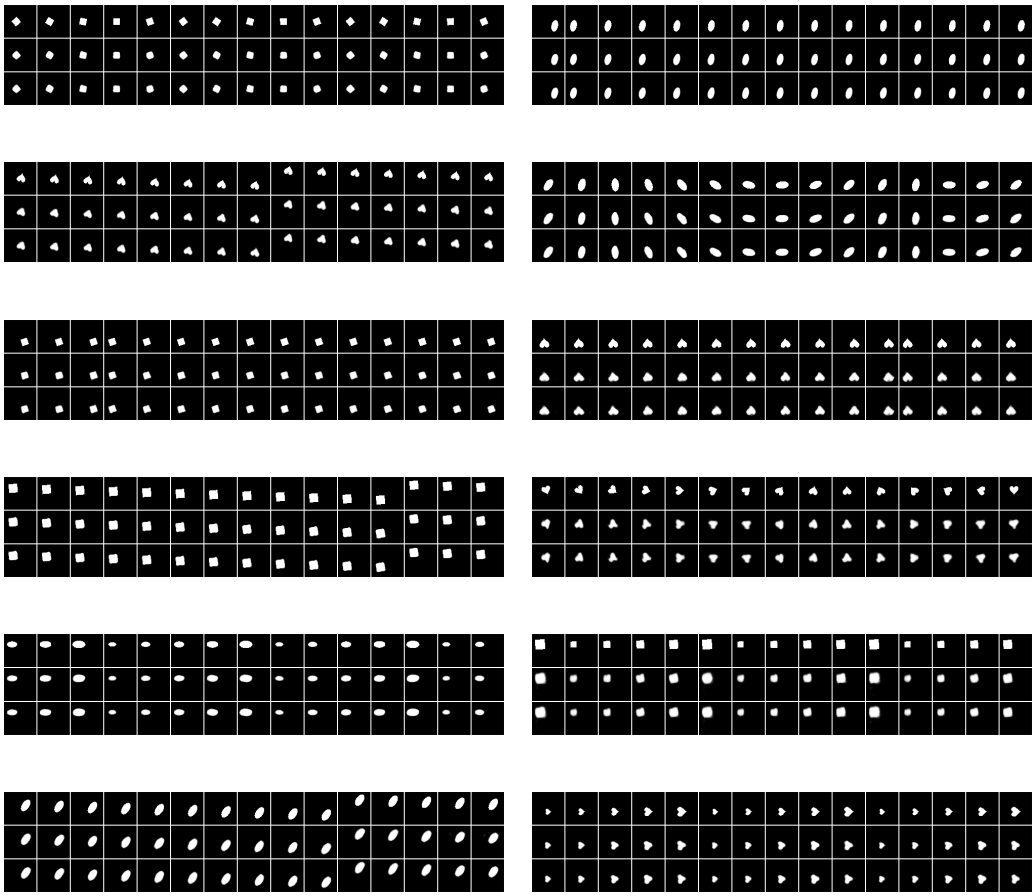


Figure B.12: dSprites TVAE $L = \frac{1}{2}S$, $K = 1$

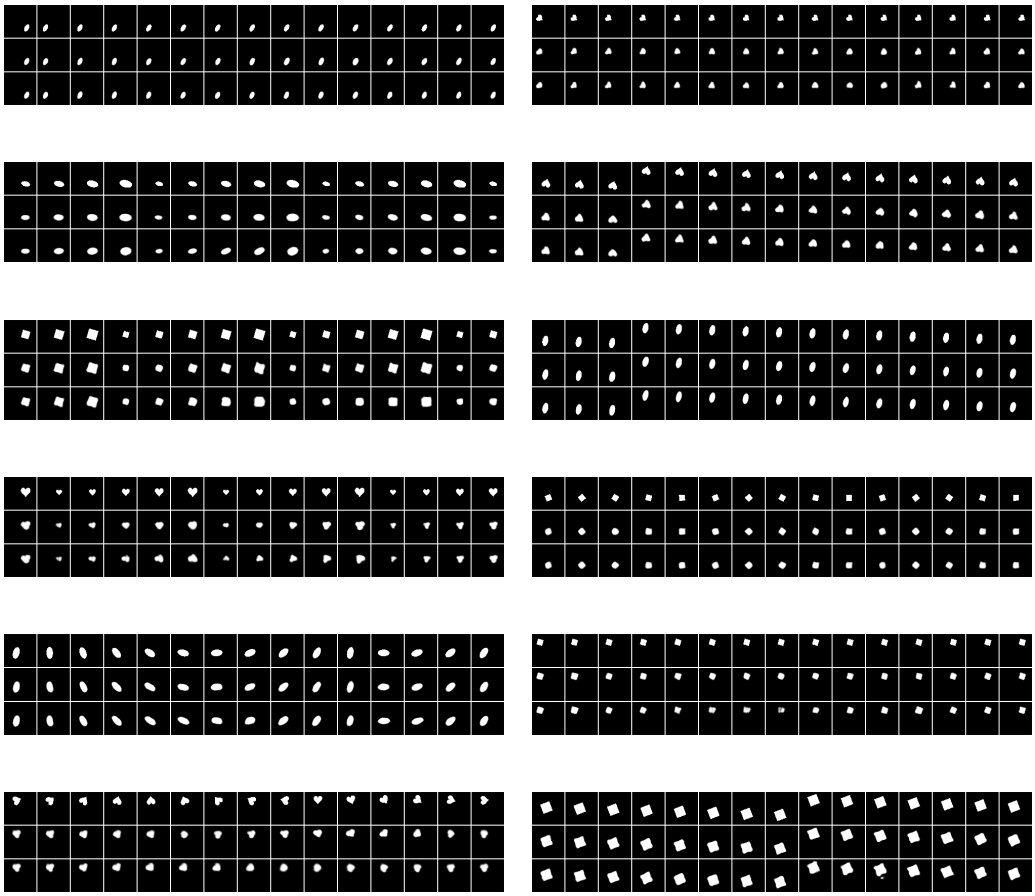


Figure B.13: dSprites TVAE $L = \frac{1}{3}S$, $K = 1$

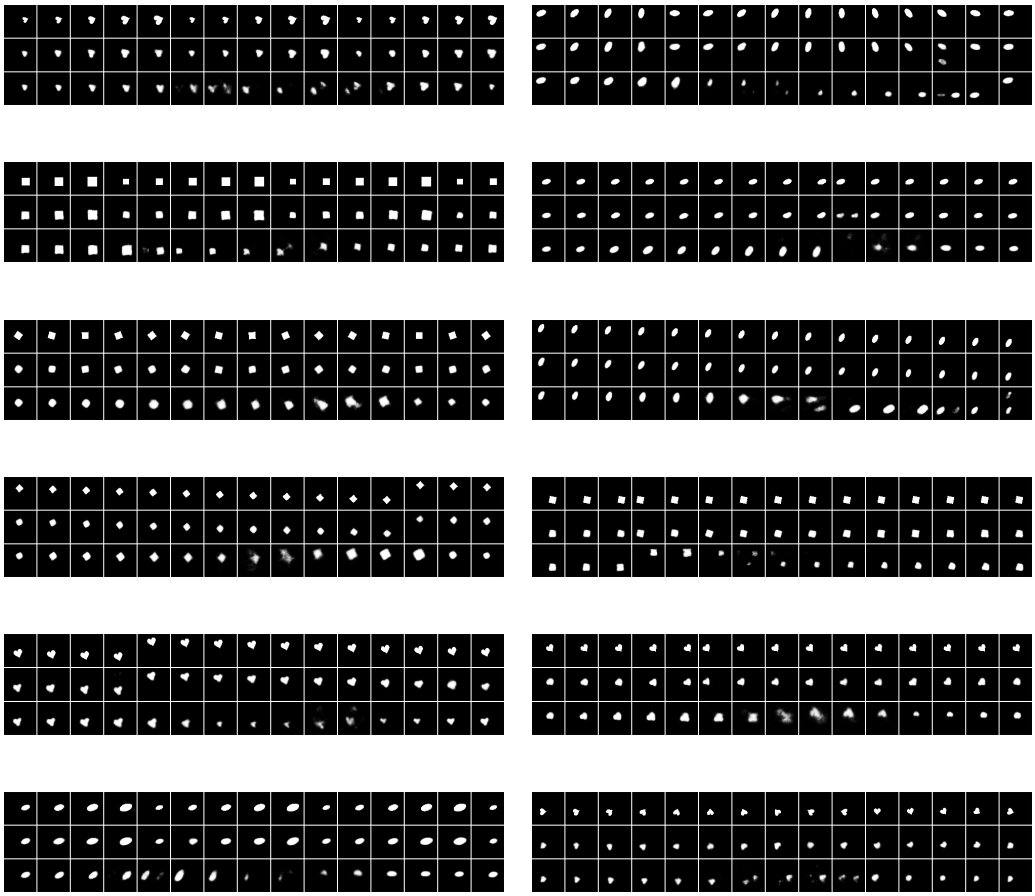


Figure B.14: dSprites TVAE $L = \frac{1}{6}S$, $K = 1$

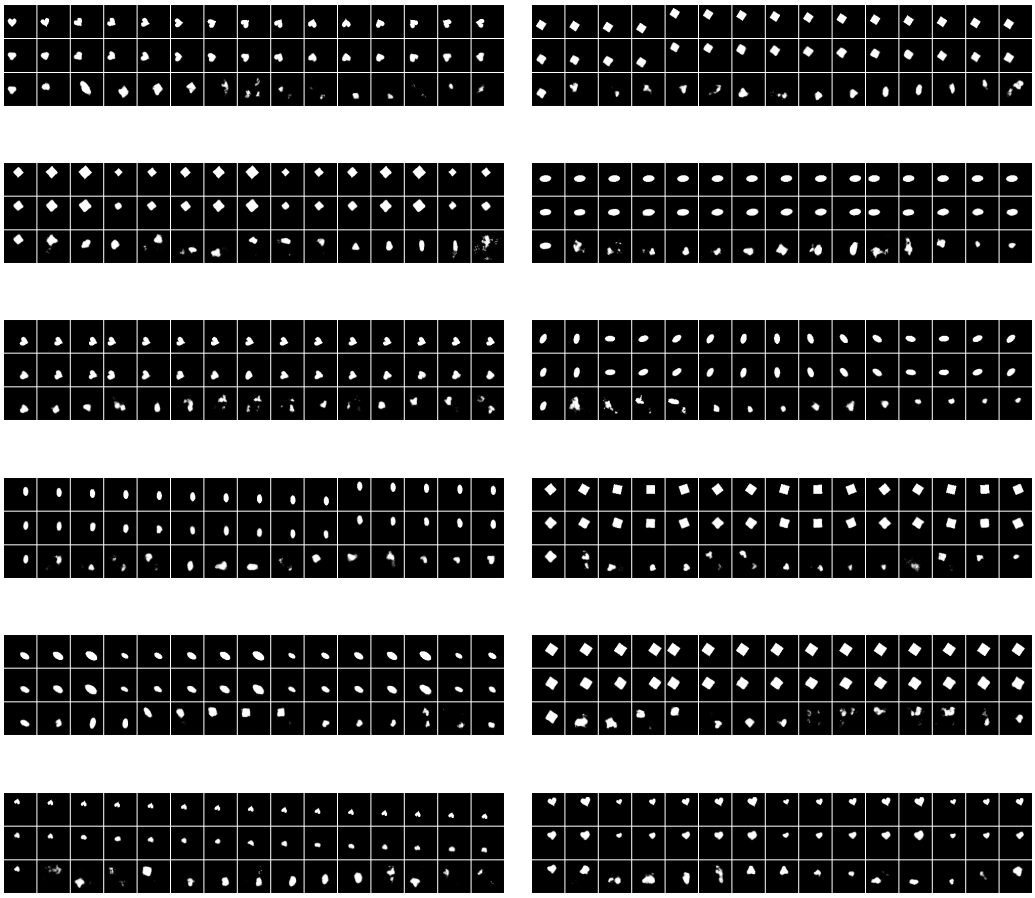


Figure B.15: dSprites TVAE $L = 0$, $K = 3$

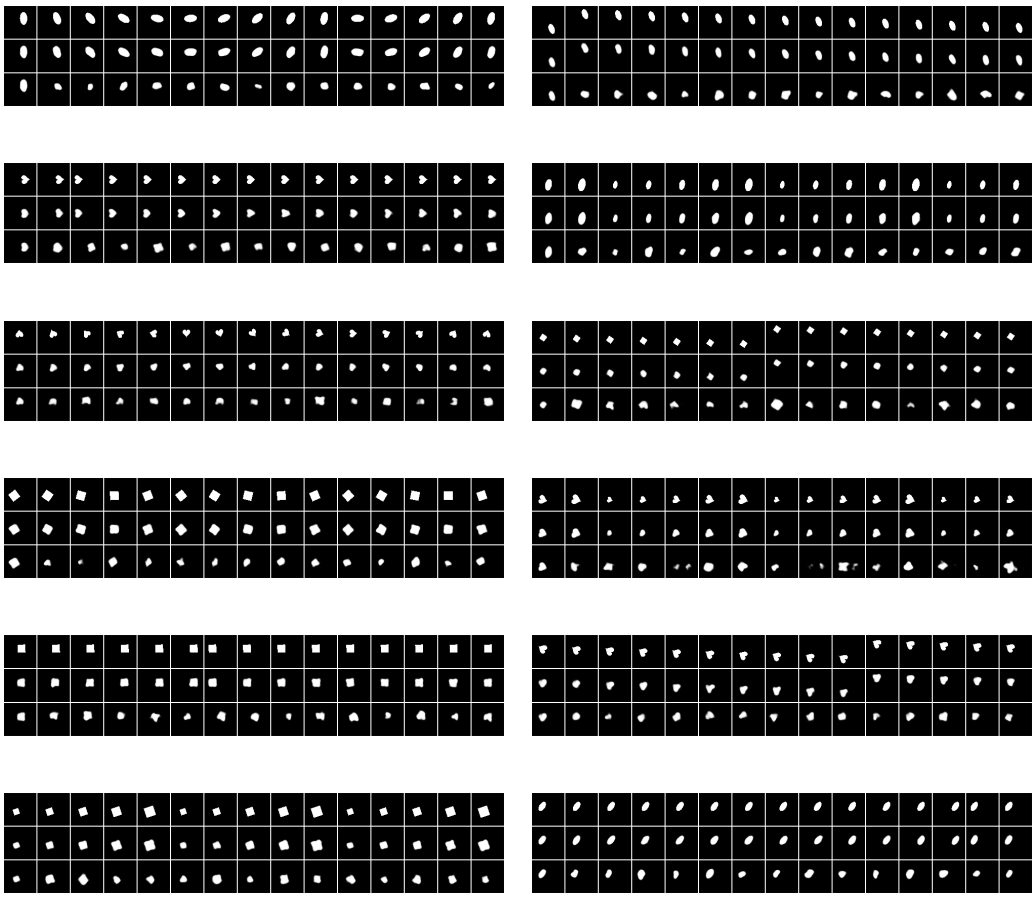


Figure B.16: dSprites BubbleVAE $L = \frac{1}{3}S$, $K = 2L$. We see the capsule traversals for the BubbleVAE produce only relatively minor transformations in the observation space (e.g. shape or rotation change, but position appears constant). This reinforces the intuition that models with stationary temporal coherence are likely to learn invariant capsule representations.

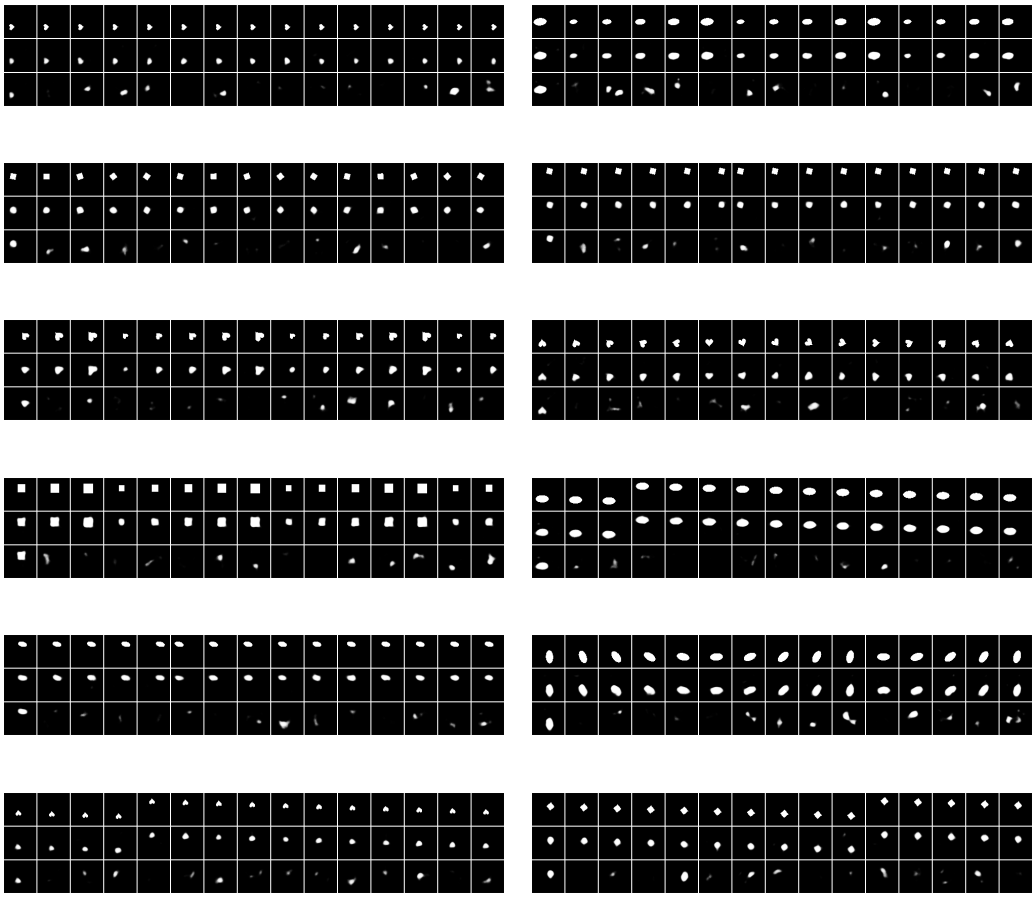


Figure B.17: dSprites VAE $L = 0$, $K = 1$

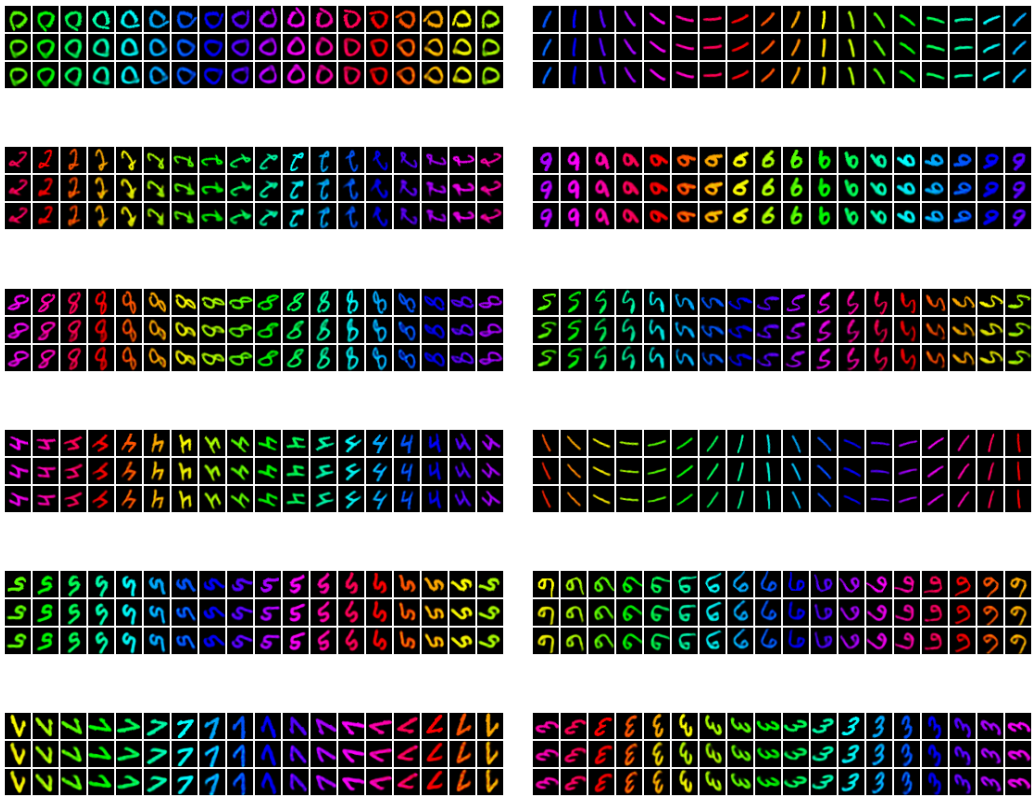


Figure B.18: Combined Color & Rotation MNIST TVAE $L = \frac{13}{36}S$, $K = 3$. We see these generated sequences are slightly more accurate than those in Figure B.2. This is to be expected since the model in this figure is trained explicitly on combinations of transformations, whereas the model in Figure B.2 was trained on transformations in isolation, and tested on combinations to explore its generalization.

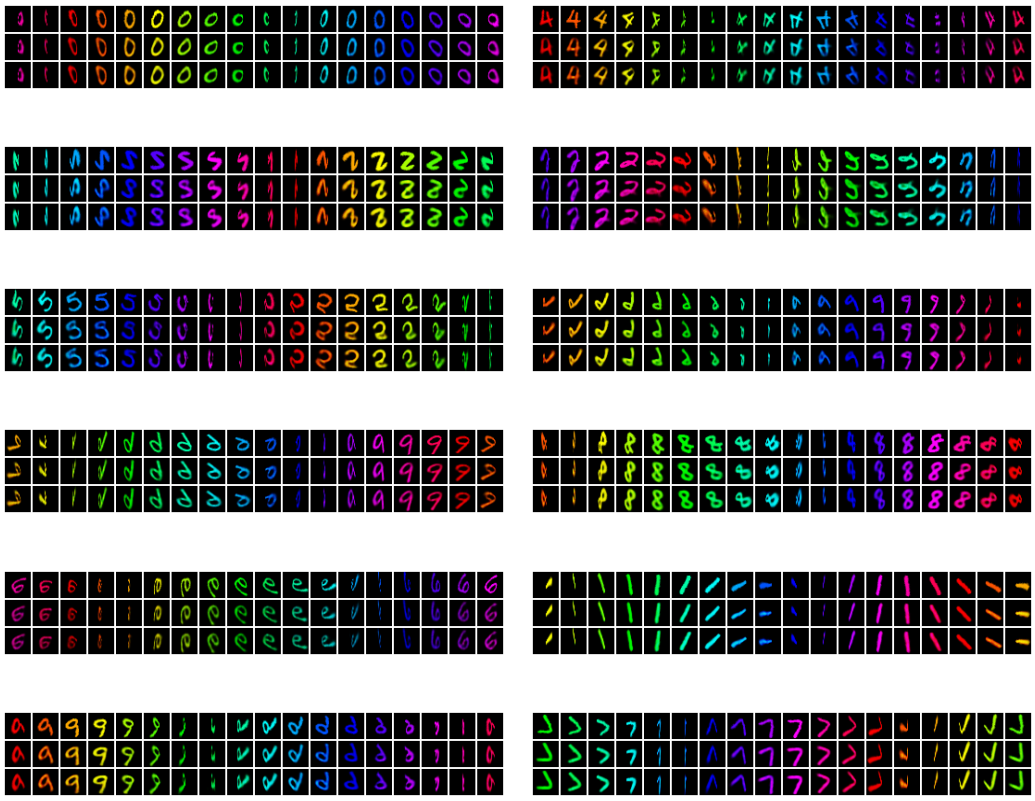


Figure B.19: Combined Color & Perspective MNIST TVAE $L = \frac{13}{36}S$, $K = 3$. We see the TVAE is able to additionally learn combinations of complex transformations (like out-of-plane rotation) without any changes to the training procedure other than a change of dataset.

Appendix C

CHAPTER VI APPENDIX

C.1. Experiment Details

Videos of traveling waves and code to reproduce all experiments in the paper can be found at the following github repository: `https://github.com/akandykeller/NeuralWaveMachines`.

The code is built as extensions of three existing public repositories, allowing us to reproduce all baseline results from the original authors' code. Specifically, we make use: (I) The coRNN repository (`https://github.com/tk-rusch/coRNN`) for the supervised sequence experiments, (II) The Topographic VAE repository (`https://github.com/akandykeller/TopographicVAE/`) for the rotating MNIST experiments, and (III) The DeepMind Physics Inspired Models repository (`https://github.com/deepmind/deepmind-research/tree/master/physics_inspired_models`) for the Hamiltonian Dynamics Suite Experiments.

Sequence Classification

The efficiency experiments from Section 6.4 were performed by modifying the published code for the original coRNN (T. Konstantin Rusch and Mishra, 2021a) to incorporate the local connectivity constraints outlined in the main text. All hyperparameters were thus set to the defaults in the published code which matched the optimal hyperparameters stated by the authors to be found from a grid search on each dataset independently. The baseline coRNN values in Table C.2 are thus simply from re-running the original authors code, and we observe similar values to those published in (T. Konstantin Rusch and Mishra, 2021a). We acknowledge that running a separate grid search for the NWM models may be beneficial to their performance but we were unable to do so due to time and computational constraints and thus leave this to future work. In practice, we found the original coRNN parameters worked well enough to give an initial intuition for the relative performance of the NWM.

For the NWM, the topology of the hidden state was defined to be a regular square 2D grid with side lengths equal to square root of the default hidden state size (or the integer floor of the square root for non-perfect-square values). Each neuron was

defined to be connected to its immediate surrounding 8 cells in the grid, in addition to a self-connection. The boundary conditions of the topology were defined to be periodic (implemented through circular padding) such that the global topology was that of a 2-dimensional torus. The recurrent local coupling parameters were shared over all spatial locations of the grid, allowing the above local connectivity to be implemented as a periodic convolution with a kernel of size 3×3. We noted that increasing the number of channels in the convolutional layers dramatically improved performance, and thus for the NWM models in Table C.2 we use 16 channels in the hidden state. This yeilded a parameter count computation of: $\#\theta = 1 \times 256 \times 16 + 16 \times 16 \times 3 \times 3 \times 2 + 256 \times 16 \times 10 = 49,664$.

Rotating MNIST and Sine Waves

The experiments on measuring spatiotemporal structure using the MNIST and simple sine waves datasets were performed by modifying the published code for the Topographic VAE (T. A. Keller and Max Welling, 2021b) to introduce our proposed NWM in place of the 'shifting temporal coherence' construction of the topographic Student's-T variable in the original paper. To achieve this, the encoder and decoder (f_θ & g_θ) were implemented as a variational autoencoder (Kingma and Max Welling, 2014) with a standard Gaussian prior and Bernoulli distribution for the likelihood of the data. Practically, this was achieved by setting the output dimensionality of the encoder f_θ to twice the hidden state dimensionality, defining half of the outputs as the posterior mean μ_θ, and the second half as the log of the posterior variance σ_θ. We additionally found that applying Layer Normalization (J. L. Ba et al., 2016) (denoted LN) to the output of the encoder helped increase convergence speed. Explicitly, the model can thus be described as:

$$\mathbf{z}_{t+1} \sim q_\theta(\mathbf{z}_{t+1}|\mathbf{u}_{t+1}) = \mathcal{N}\big(\mathbf{z}_{t+1}; \mu_\theta(\mathbf{u}_{t+1}), \sigma_\theta(\mathbf{u}_{t+1})\mathbf{I}\big), \qquad \bar{\mathbf{z}}_{t+1} = \mathrm{LN}(\mathbf{z}_{t+1}) \tag{C.1}$$

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t\big(\sigma\left(\mathbf{w}_x \star \mathbf{x}_t + \mathbf{w}_{\dot{x}} \star \mathbf{v}_t + \mathbf{V}\bar{\mathbf{z}}_{t+1} + \mathbf{b}\right) - \gamma\mathbf{x}_t - \alpha\mathbf{v}_t\big) \tag{C.2}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t\left(\mathbf{v}_{t+1}\right) \tag{C.3}$$

$$p_\theta(\mathbf{u}_{t+2}|g_\theta(\mathbf{x}_{t+1})) = \mathrm{Bernoilli}(\mathbf{u}_{t+2}; g_\theta(\mathbf{x}_{t+1})) \tag{C.4}$$

Where the objective is then computed by averaging the evidence lower bound (ELBO) over the length of the sequence:

$$\mathcal{L}(\mathbf{u}_{1:T}; \boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mathbf{z}_t \sim q_\theta(\mathbf{z}_t|\mathbf{u}_t)}\big(\log p_\theta(\mathbf{u}_{t+1}|g_\theta(\mathbf{x}_t)) - D_{KL}[q_\theta(\mathbf{z}_t|\mathbf{u}_t)||p_{\mathbf{Z}}(\mathbf{z}_t)]\big) \tag{C.5}$$

The initial conditions for the NWM were then given by simply setting the initial position equal to the first encoder output, and the initial velocity to zero, i.e. $\mathbf{x}_0 = \bar{\mathbf{z}}_0$ & $\mathbf{v}_0 = \mathbf{0}$. Although we did not test the MNIST experiments with a deterministic autoencoder, we note that traveling waves can also clearly be seen in the hidden states of the deterministic models presented in Sections 6.3 and 6.4 (as visualized in Figure 6.2 and the supplementary material), implying that the variational formulation is not necessary for the emergence of traveling waves.

For the experiment depicted in Figure 7.2 of Section 6.4, we used a simple linear encoder and decoder, and a hidden state dimensionality of 1296 reshaped into a 2D grid of shape 36×36. As in the rest of the paper, our topographic connectivity was implemented using a convolutional kernel of shape 3×3 shared over all elements of the grid, with circular padding to enforce periodic boundary conditions on the grid. For training, we presented the model with length 18 sequences of MNIST digits rotating at 20 degrees per step (thus completing a full period per training sequence). At test time, to create the visualization in Figure 7.2, we increased the sequence length to 72 elements (or four periods) and visualize a portion of the final period, allowing the system to reach a steady state of wave activity for better visualization. We see that despite not being trained on such long sequences, the NWM is able to generalize and maintain wave activity. For computing the generalized phase, we set use a 4-th order butterworth bandpass filter with bounds set at 0.2 and 0.4 of the Nyquist frequency. As hyperparamters for training, we used standard SGD with momentum of 0.9, a learning rate of 2.5×10^{-4}, and a batch size of 128 for 50 epochs. Following the suggestion outlined in (T. Konstantin Rusch and Mishra, 2021a), we allowed the parameters γ, α, & Δt to be learned during training by initializing them to $\Delta t = \sigma^{-1}(0.125) = -1.95$, $\gamma = 1.0$, & $\alpha = 0.5$ and then applying appropriate activation functions to keep them within the desired bounds (e.g. sigmoid, ReLU, & ReLU respectively). These hyperparameters and initalization values were determined by implementing a simple toy version of the model with random data and random weights and manually altering parameters to determine the ranges for which coherent wave dynamics were likely to emerge. We note that the properties of the emergent waves appear qualitatively different for different random initalizations of the model. Specifically the wavelength and velocity of the waves appears to vary greatly from run-to-run. We show a few of these different learned dynamics in the additional results section below.

For the experiment depicted in Figure 6.5 of Section 6.4, we used a 3-layer Multi-

Layer Perceptron (MLP) for the both encoder and decoder, and a hidden state of dimensionality 1296 reshaped into a set of 24 disjoint 1-D tori (circles) each composed of 54 neurons. We implemented topographic coupling between the immediate neighbors on each circle via a 1-dimensional convolutional kernel of size 3 with circular padding. We then implemented the uni-directionality constraint outlined in the main text be masking the first two elements of the kernel to 0, yielding a kernel with a single trainable parameter explicitly connecting each neuron with its neighbor directly to one side. For training, the dataset and hyperparameters all remained the same as in Figure 7.2 described above, however the batch size was reduced to 8 for quicker evaluation. We found that additionally adding another layer normalization layer between recurrent steps improved the consistency of the learned waves and thus allowed us to simulate them more accurately at test time. Explicitly this amounted to modifying Equation C.3 to: $\mathbf{x}_{t+1} = \text{LN}\big(\mathbf{x}_t + \Delta t\,(\mathbf{v}_{t+1})\big)$. Furthermore, to ensure consistency of waves across each circular subspace separately, we shared the bias vector $\mathbf{b}$ across each subspace. To induce a traveling wave in the hidden state of the network and thereby generate the transformation sequence shown in the bottom row of the figure, we first encode the input sequence (shown in the top row), using the equations outlined in this section. We take the final hidden state of the network ($\mathbf{x}_T$) as the initial state from which we begin the wave propagation. Then, across each 1-D circular subspace of the hidden state, we update the values of the hidden state based on the 1-D 1-way wave equation $y(x,t) = f(x - vt)$ for a velocity $v = 1$ for time $t = 1$ to 18. Written in terms of the hidden state $\mathbf{x}_t$, we can effectively propagate waves backwards through the hidden state by moving activation from one spatial location l to a location shifted by $v\Delta t$: $\mathbf{x}_T(l) \rightarrow \mathbf{x}_T(l - v\Delta t)$. Practically, this amounts to sequentially circularly shifting the hidden state activation across each circular subspace as depicted in Figure 6.5.

Hamiltonian Dynamics Suite

The experiments in Section 6.4 were performed using the DeepMind Physics Inspired Models and Hamiltonian Dynamics Suite, implemented in JAX, as a starting point. All values reported for the baselines (HGN++, AR, and ODE [TR]) were thus obtained by re-running the original code with the hyperparameters stated in (Botev et al., 2021). Specifically, for the HGN++, we trained the model both forwards and backwards in time, including over the inference steps, with a final beta value of 0.1 in the ELBO. For the AR model, we used an LSTM with all other paramters default.

For the ODE, we used the default parameters with forwards and backwards training, again including inference steps. The only change to the default hyperparamters for all three models was to reduce the batch size to 8 per GPU (thus 32 total per iteration) to fit on our GPUs.

The coRNN and NWM architectures were added as extensions to the auto-regressive model already implemented in library. They thus made use of all the same default hyperparameters, with the only changed values being the aforementioned reduced batch size, an increased number of inference steps (31), an increased number of target steps (60), and an increased hidden state size (23×23). The increased number of inference and target steps was found useful to improve performance on more chaotic tasks such as the pendulum where the accuracy of the initial state is hugely important to the model forecasting performance. Additionally, we note that these values are within the values searched by the grid search of the authors in (Botev et al., 2021) making their use here for comparison relatively fair. The size of the hidden state was picked as the largest which fit in our GPU memory across all devices. The values of α, γ, and Δt were initialized to the same values as the MNIST experiments described above, and were again allowed to be updated during training simultaneously with the other model parameters. For the 2D NWM, the hidden state topology was again defined to be a 2D torus of size (23×23) implemented through periodic convolution with a 3×3 kernel. The 1D NWM topology was similarly composed of 23 disjoint 1D circles each with 23 neurons, again implemented with periodic convolution with a 1×3 kernel. The coRNN and NWM models additionally used a separate initial condition network to initialize $\mathbf{x}_0$ and $\mathbf{v}_0$. This network was implemented as a GRU with a hidden state of size $2 \times 23 \times 23$ which ran backwards over the inference sequence (length 31) first embedded with the model encoder f_θ. The final hidden state of the model was then split in half and taken to initialize the inital positions and velocities of the coRNN & NWMs.

All models make use of the same deep convolutional encoder with ReLU activations and a similarly deep convolutional spatial broadcast decoder as in the original work. They were similarly all trained for 500,000 iterations to match the original work.

Hardware Details

All models were run on a cluster across roughly 8 NVIDIA GeForce 1080Ti GPUs, 8 NVIDIA GeForce 980Ti GPUs, and 8 NVIDIA Titan X Gpus. Each model in Table 6.1 thus required roughly 6-8 GPU days to train to the final number of iterations.

C.2. Analytical Treatment of Neural Wave Machines

In this section we extend the analytical treatment of Neural Wave Machines, verifying that the model does indeed inherit many of the same beneficial bounds on hidden state and gradient magnitudes as the original coRNN, as stated in the main text. Specifically, by carefully reviewing the proofs for Proposition 3.1 (bounded hidden state energy) and Proposition 3.2 (bounded hidden state gradients) of T. Konstantin Rusch and Mishra (2021a), it can be shown that the Neural Wave Machine satisfies the conditions necessary for these bounds to similarly hold with minor modifications. At a high level, the intuition for why these bounds hold is that our convolutional parameterization of the coupling matrices does not change the theoretical bounds on the infinity norm of the weights, the crucial element necessary for bounding these quantities (e.g. see equation (13) of T. Konstantin Rusch and Mishra (2021a)). In the following, we detail each of these bounds more precisely.

Bounds on Hidden State Energy

Identically following the proof of Proposition 3.1, from Section E.1 of T. Konstantin Rusch and Mishra (2021a), defining the total energy of our model's hidden state as $\mathbf{x}_n^T\mathbf{x}_n + \mathbf{v}_n^T\mathbf{v}_n$, it can be seen this value is bounded at time-step n, and with hidden state size m, as:

$$\mathbf{x}_n^T\mathbf{x}_n + \mathbf{v}_n^T\mathbf{v}_n \leq \mathbf{x}_0^T\mathbf{x}_0 + \mathbf{v}_0^T\mathbf{v}_0 + nm\Delta t$$

We see that this bound does not change from the original work as the derivation is not dependent on the parameterization of the coupling matrices $\mathbf{W}, \mathcal{W}$. Furthermore, this bound applies equally in the case when we have non-zero initial conditions (as through our initial condition network).

Sensitivity to Inputs

From Section E.2, Proposition E.1, of (T. Konstantin Rusch and Mishra, 2021a), it can be seen that the NWM also inherits a bound on how much differences in inputs are able to change the hidden state. Specifically, since the activation function we use is tanh, our bound is identical. This is the theoretical justification for our comment regarding the NWM's apparent inability to model chaotic dynamics (which we expand on in Appendix C.3).

Bounds on Hidden State Gradient

From Section E.3, following the proof of Proposition 3.2, of T. Konstantin Rusch and Mishra (2021a), we see that, assuming $\alpha = \gamma = 1$, we can again derive bounds on the gradient of the loss with respect to the model parameters. Specifically, the outline of the proof is nearly identical, with only equation (28) being modified to reflect the fact that our parameters are now shared over all spatial locations (due to the convolution). In detail, the matrix $\mathbf{Z}^{i,j}_{m,\bar{m}}$ no longer only has a single non-zero value, but instead m non-zero values equal to $\sigma'(\mathbf{A}_{k-1})_i$ (for an m sized hidden state). We see that when this matrix is then multiplied with each vector $(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}, \mathbf{u}_k)$, using the bound $||\mathbf{Z}^{i,j}_{m,\bar{m}}(\mathbf{A}_{k-1})||_\infty \leq 1$, the upper bounds in equation (29) change from $||\mathbf{x}_{k-1}||_\infty$, $||\mathbf{v}_{k-1}||_\infty$, $||\mathbf{u}_k||_\infty$ to $m||\mathbf{x}_{k-1}||_\infty$, $m||\mathbf{v}_{k-1}||_\infty$, $m||\mathbf{u}_k||_\infty$. Carrying these extra factors of m through the rest of the proof, we arrive at the following final bound on the gradient of the loss function ξ with respect to any parameter θ:

$$\left|\frac{\partial \xi}{\partial \theta}\right| \leq \frac{3}{2}(m + \bar{X}m^{3/2})$$

where $\bar{X} = \max_n ||\bar{x}_n||_\infty$.

Assumptions

As with the proofs for the coRNN, the same assumptions are necessary for the bounds to hold. Specifically, it is assumed that Δt is chosen such that:

$$\max\left(\frac{\Delta t(1 + ||\mathbf{W}||_\infty)}{1 + \Delta t}, \frac{\Delta t||\mathcal{W}||_\infty}{1 + \Delta t}\right) \leq \Delta t^r, \quad \frac{1}{2} \leq r \leq 1 \tag{C.6}$$

Since this assumption is indeed satisfied throughout training for the original coRNN, we assume that it is likely satisfied with the NWM as well. Intuitively, we find no reason to believe that changing the fully connected matrices $\mathbf{W}$ & $\mathcal{W}$ to convolutional matrices will have the necessary order-of-magnitude impact on the infinity norm of the weight matrices necessary to invalidate this assumption. In preliminary experiments on sMNIST we also find this intuition to hold. Specifically, for the optimal value of $\Delta t = 0.042$, and $r = \frac{1}{2}$, we see that the maximum over training of the quantity of interest (Equation C.6) is actually lower for the NWM than the coRNN (0.157 vs. 0.188) with both being lower than the limit (0.205).

C.3. Extended Results

Impact of Δt parameter

In this section we include an additional preliminary analysis to measure the impact of changing the Δt parameter. In practice, we see that the parameter has an impact not only on the numerical integration, but also on the speed at which the network's hidden state is able to update. Therefore, similar to prior work with the coRNN, we find it best to treat this parameter as a hyperparameter and tune it in addition to the other hypterparameters. In the table below, we show the results of our model on sMNIST for a range of Δt values:

Table C.1: Test accuracy on the sMNIST dataset for a range of Δt values.

Δt	0.001	0.1	0.042	0.15	0.30	0.45
Test Accuracy	87.7	90.6	98.4	97.5	89.8	NaN

We see that a moderate value of Δt is optimal, while too large causes divergence (perhaps due to excessive discretization errors) and too small disrupts information processing in the RNN.

Additional Efficient Sequence Modeling Results

In this section we include additional results comparing the coRNN and NWM on different sequence modeling tasks. Specifically, we show model performance on the long-sequence addition task initially introduced by Hochreiter and Schmidhuber (1997), and the IMDB sentiment classification task (T. Konstantin Rusch and Mishra, 2021a). On both datasets we see that the NWM achieves comparable performance to the coRNN while requiring significantly fewer parameters, in line with results on the sMNIIST and psMNIST datasets.

Additional Hamiltonian Dynamics Results

In this section we include an alternative metric for measuring model forecasting performance on the Hamiltonian Dynamics Suite. Specifically in Table C.3, we report the 'Valid Prediction Time' as reported in prior work (Botev et al., 2021), defined as the number of time steps into the future the models are able to accurately predict the dynamics of the system with reconstruction error under a predefined threshold (MSE < 0.025). Given the high variance of the VPT value from batch-to-batch, the

Table C.2: Test accuracy on additional sequence modeling benchmarks including the long-sequence Addition task from Hochreiter and Schmidhuber (1997), and the IMDB sentiment classification task. All results are mean ± std. over 3 random initalizations. We see similar results to those shown in Table C.2, the NWM achieves comparable performance while requiring significantly fewer parameters.

	Adding Task		IMDB	
	Accuracy	$\#\theta$	Accuracy	$\#\theta$
coRNN	0.0035 ± 0.01	131k	86.4 ± 0.2	46k
NWM	0.0046 ± 0.0016	<1k	86.1 ± 0.3	13k

values reported in Table C.3 are computed as the mean and standard deviation of the VPT over the final 5 evaluation iterations. We see that the values roughly agree with those reported in (Botev et al., 2021), however certain discrepancies may still appear due to the fact that the authors of (Botev et al., 2021) only report the range of the grid search they performed but not the actual hyperparameter values of their best performing models. Further, we see that the ranking of model performance under this metric is quite noisy due to the high variance of the metric. We therefore urge future work to consider alternative benchmarks and metrics for evaluating the forecasting performance of such models.

Table C.3: Valid Prediction Time 'VPT' (± std.) on the Hamiltonian Dynamics Benchmark. We highlight in bold results which fall within one standard deviation of the best performing model. We see that the VPT metric has large standard deviation owing to the reliance on an arbitrary threshold of image-space similarity, however the NWM models still perform favorably compared with existing state of the art.

	AR	HGN++	ODE [TR]	coRNN	NWM 2D	NWM 1D
Spring	302 (63)	**447 (0)**	**430 (26)**	375 (14)	311.8 (27)	**431 (24)**
Pendulum	3 (4)	105 (21)	**212 (65)**	**179 (91)**	155.1 (24)	**174 (65)**
Two Body	263 (92)	**444 (3)**	**439 (11)**	**431 (40)**	**413 (53)**	**420 (27)**
Pennies	118 (25)	79 (6)	**164 (14)**	**165 (23)**	**141 (37)**	**163 (9)**
Double Pendulum	0 (0)	11 (5)	**22 (7)**	3 (1)	9 (9)	**10 (8)**

On Modeling Chaotic Dynamics

In this section, we include an extended evaluation to investigate the apparent inability of the NWM to model more chaotic dynamics such as the double pendulum task. To do this, we perform an analogous experiment to that reported in Appendix A of the original coRNN work (T. Konstantin Rusch and Mishra, 2021a). Specifically, we measure the ability of our model to predict the state of a system at a fixed 25-time

steps ahead for a Lorentz '96 attractor ($x'_j = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$). Here, F is an external force which controls how chaotic the trajectories are, where $F = 8$ corresponds to a highly caotic trajectory and $F < 1$ is significantly less chaotic. Ultimately, we see that, similar to the original coRNN work, the LSTM performs significantly better than the NWM in the chaotic regime, providing empirical evidence for the theoretical claim that the coupled oscillator networks are unable to model chaotic dynamics.

Table C.4: Test Mean Squared Error of an LSTM and NWM when forecasting the Lorentz '96 attractor. We see that the NWM performs better in the non-chaotic regime ($F = 0.9$), while in chaotic regime ($F = 8$) the LSTM performs significantly better.

Model	$F = 0.9$	$F = 8.0$
LSTM	5.2×10^{-3}	1.9×10^{-2}
NWM	2.4×10^{-3}	4.8×10^{-2}

On the Formation of Orientation Maps

Although there is significant prior work which can give intuition as to why the smooth orientation selectivity maps of Figure 6.3 may arise from our model, we believe we are the first to demonstrate a system which actually learns these types of maps from data in the service of sequence modeling. At the highest level, the intuition for the mechanism behind these maps can be seen to come from the combination of phase-synchrony of coupled oscillator systems, and the necessity to model temporally correlated transformations. Extensive prior work on so-called 'phase-reduced' Kuramoto models demonstrates the emergence of complex spatiotemporal patterns such as plane waves, spirals, and pinwheel lattices. Examples include early work from B. Ermentrout et al. (1970) (Figure 6), showing various steady state phase relationships in the solutions of the locally coupled oscillator dynamics. Similarly, more recent works, (S.-O. Jeong et al., 2002) (Figs. 3 & 4) and (Breakspear et al., 2010) (Figs. 5 & 6) have studied how this phase-locking can vary for different types of chosen couplings. Given that these phase-reduced systems are theoretical approximations to the more flexible (non-reduced) oscillator dynamics implemented in the NWM, it makes sense that we also see these types of phase relationships (e.g. Fig. 7.2 of the main text). When such complex phase-synchrony is combined with the task of sequence modeling, the synchrony can be seen to essentially be inducing local correlations between neurons for each time-step. Thus, when the training set contains input at a variety of different angles, and the model is required to represent these over time, the intuition follows that there will be spatially-

smooth orientation selectivity corresponding to these induced correlations. In Figure C.1 we provide some quantitative measurements which align with this intuition. Specifically, the figure shows the instantaneous phase measurement of each neuron (right) next to the orientation selectivity of the same neurons (left). As can be seen, there is a rough correlation between phase values and orientation selectivity, with unexplained variance likely arising due to computing the depicted instantaneous phase values from a single training example, while selectivity measurements are computed over an entire dataset. Furthermore, in Figure C.2 we show how different hyperparameters affect the size of the resulting learned orientation columns. We see that both the wavelength of the training dataset (λ^{train} of sine waves) and the kernel size (size($\mathbf{w}_z$)) have a direct increasing relationship with the size of the learned orientation columns, suggesting these parameters could be tuned to better fit observations from neuroscience.

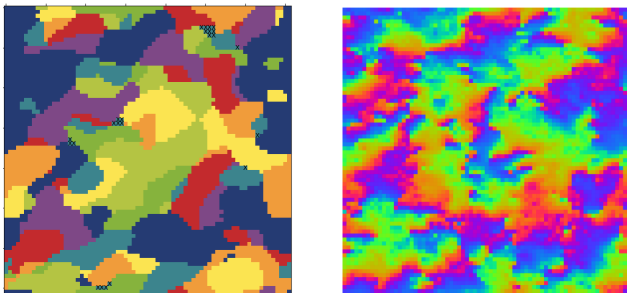


Figure C.1: Orientation selectivity (left) and instantaneous phase at a random sequence element (right) for a model trained on the sine waves dataset. We see that the phase synchrony across the neurons is roughly in alignment with the orientation selectivity, supporting the hypothesis that this is one of the primary mechanisms for topographic organization in the NWM.

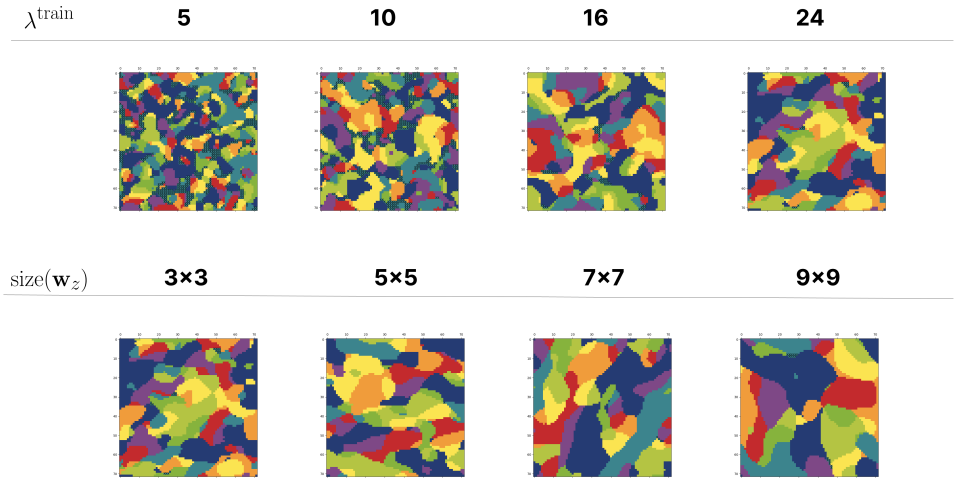


Figure C.2: Orientation selectivity maps as a function of training dataset wavelength (λ^{train}), and kernel size (size($\mathbf{w}_z$)).

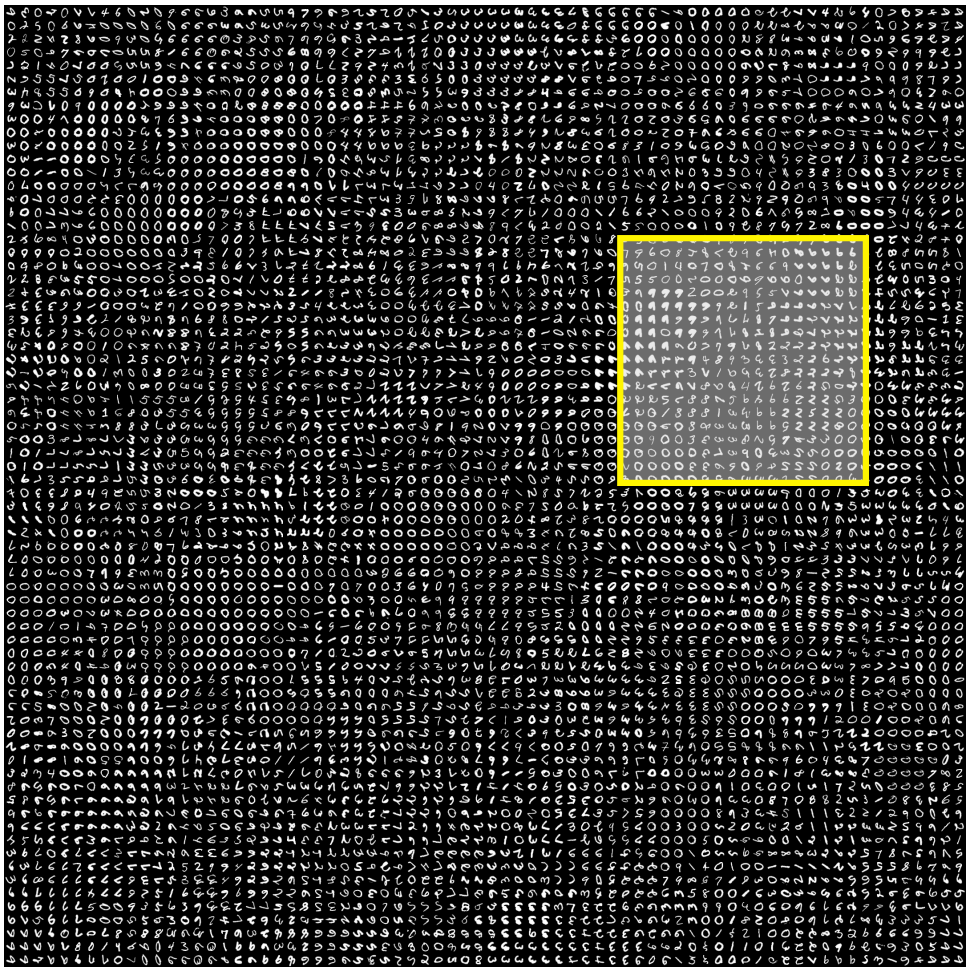


Figure C.3: Depiction of the maximum activating image for the full set of neurons in the NWM when training on Rotated MNIST. The subset depicted in Figure 6.3 is highlighted in yellow. We see that topographic organization is widespread and roughly continuous throughout the hidden state.

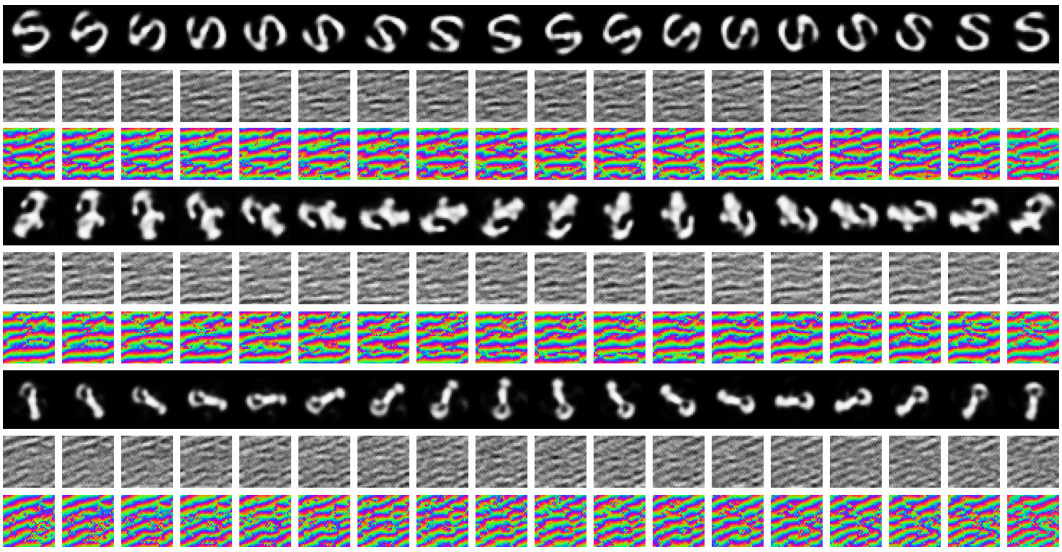


Figure C.4: Additional hidden state visualizations for the model in Figure 7.2. Reconstructions (Top), Hidden state (middle) and generalized phase (bottom), for the final 18 timesteps of the test sequence.

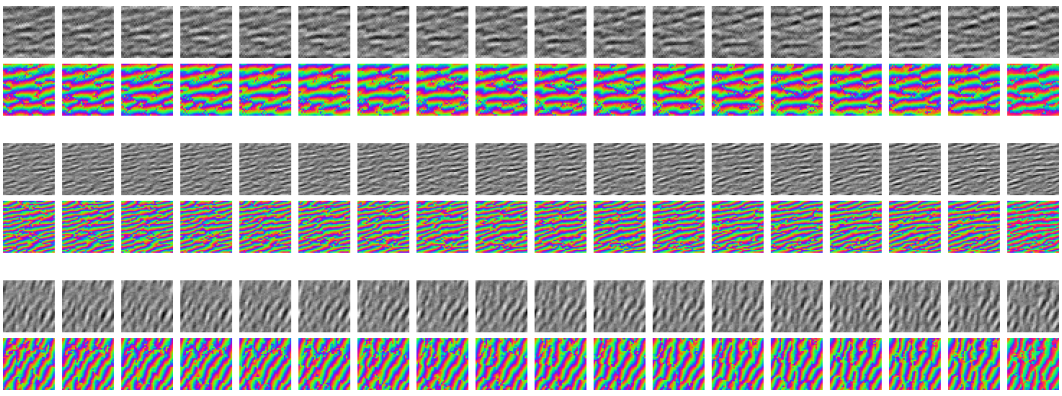


Figure C.5: Visualization of the hidden state and phase for three models identical to those in Figure 7.2, but with different random initalizations. We see that the models learn different wavelengths and velocities depending on their initialization.

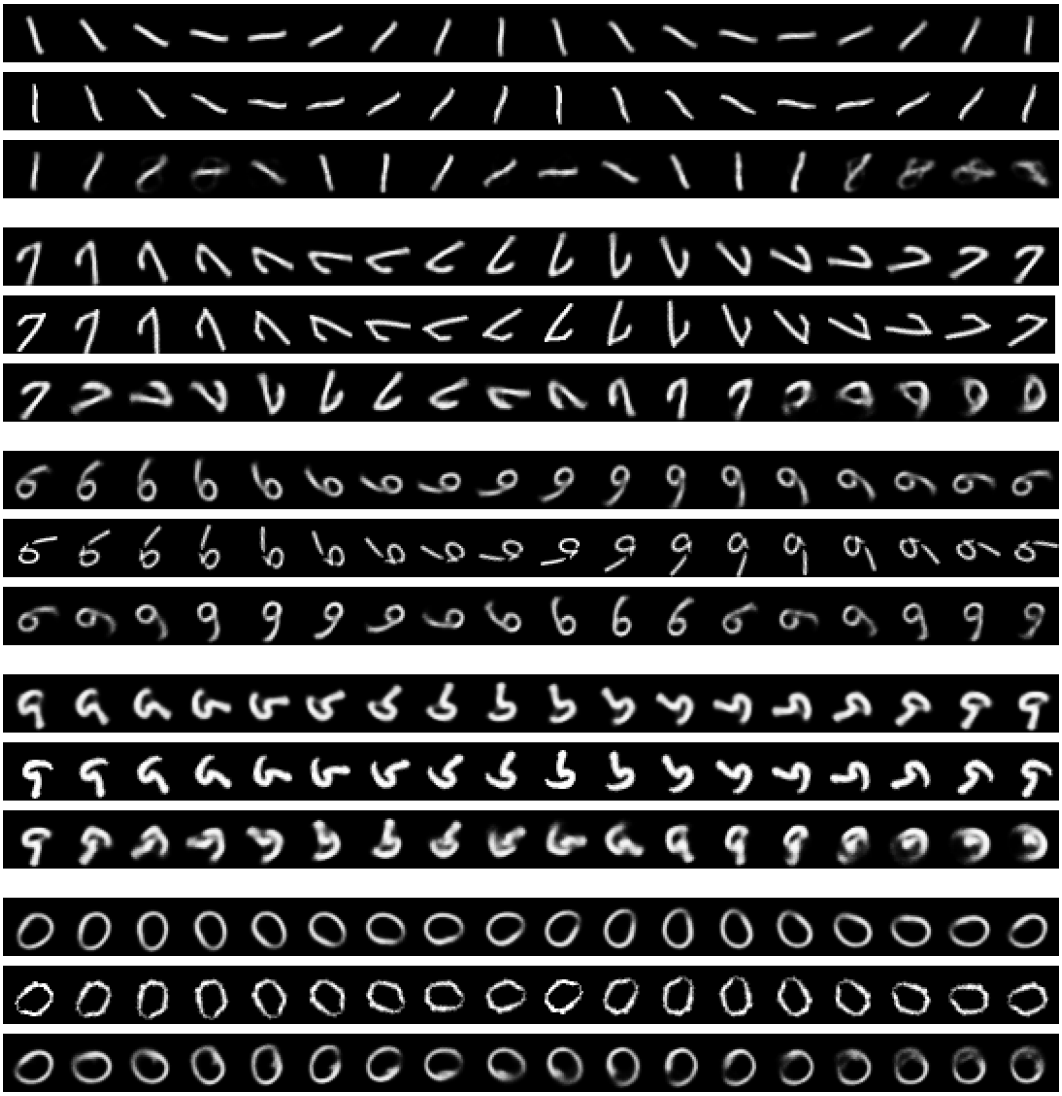


Figure C.6: Additional visualizations of reconstructions from induced wave activity in the hidden state of the 1D NWM as depicted in Figure 6.5. We show a set of random input sequences (top), the original model reconstruction (middle), and images generated by sequentially propagating the initial state backwards by an induced wave and decoding at each step (bottom). We see that, as in the main text, the assumed wave velocity of $v = 1$ is slightly faster than the actual velocity, and thus the reconstructed transformations are slightly faster than the input transformations. Because of this, we also observe that for certain examples, the induced wave reconstructions lose consistency with the input after the first period. This appears to imply that both the initial location of the wave activity matters in addition to its wave properties, and thus our model has learned to only propagate waves over parts of the feature space to optimize the capacity of the hidden state for this dataset. Finally, we observe that the induced transformations occur in reverse order due to the fact that our induced waves propagate in the reverse direction to those naturally exhibited for training examples, effectively propagating backwards in time.

Appendix D

CHAPTER VII APPENDIX

D.1. Related Work

Deep neural network architectures that exhibit some brain-like properties are related to ours: CORnet (Kubilius et al., 2019), a shallow convolutional network with added layer-wise recurrence shown to be a better match to primate neural responses; models with topographic organization, such as the TVAE (T. A. Keller and Max Welling, 2021b) and TDANN (H. Lee et al., 2020); and models of hippocampal-cortex interactions such as the PredRAE (Y. Chen et al., 2022), and the TEM (Whittington et al., 2020). Our work is unique in this space in that it is specifically focused on generating spatio-temporally synchronous activity, unlike prior work. Furthermore, we believe that our findings and approach may be complimentary to existing models, increasing their ability to model neural dynamics by inclusion of the Wave-RNN fundamentals such as locally recurrent connections and shift initalizations.

In the machine learning literature, there are a number of works which have experimented with local connectivity in recurrent neural networks. Some of the earliest examples include Neural GPUs (Kaiser and Sutskever, 2016) and Convolutional LSTM Networks (X. Shi et al., 2015). These works found that using convolutional recurrent connections could be beneficial for learning algorithms and spatial sequence modeling respectively. Unlike these works however, Chapter 7 explicitly focuses on the emergence of wave-like dynamics in the hidden state, and further studies how these dynamics impact computation. To accomplish this, we also focus on simple recurrent neural networks as opposed to the gated architectures of prior work – providing a less obfuscated signal as to the computational role of wave-like dynamics.

One line of work that is highly related in terms of application is the suite of models developed to increase the ability of recurrent neural networks to learn long time dependencies. This includes models such as Unitary RNNs (Arjovsky, Shah, et al., 2016), Orthogonal RNNs (Henaff et al., 2016), expRNNs (Lezcano-Casado and Martínez-Rubio, 2019), the chronoLSTM (Tallec and Ollivier, 2018), anti.symmetric RNNs (Chang et al., 2019), Lipschitz RNNs (Erichson et al., 2021), coRNNs (T. Konstantin Rusch and Mishra, 2021a), unicoRNNs (T. Konstantin Rusch and Mishra, 2021b), LEMs (T Konstantin Rusch et al., 2022), Recurrent Linear Units (Orvieto

et al., 2023), and Structured State Space Models (S4) (Gu, Goel, et al., 2022). Additional models with external memory may also be considered in this category such as Neural Turing Machines (Graves, Wayne, and Danihelka, 2014), the DNC (Graves, Wayne, Reynolds, et al., 2016), memory augmented neural networks (Santoro et al., 2016) and Fast-weight RNNs (J. Ba et al., 2016). Although we leverage many of the benchmarks and synthetic tasks from these works in order to test our model, we note that our work is not intended to compete with state of the art on the tasks and thus we do not compare directly with all of the above models. Instead, Chapter 7 intends to perform a rigorous empirical study of the computational implications of traveling waves in RNNs. To best perform this analysis, we find it most beneficial to compare directly with as similar of a model as possible which does not exhibit traveling waves, namely the iRNN. We do highlight, however, that despite being distinct from aforementioned models algorithmically, and arguably significantly simpler in terms of concept and implementation, the wRNN achieves highly competitive results. Finally, distinct from much of this prior work (except for the coRNN (T. Konstantin Rusch and Mishra, 2021a)), our work uniquely leverages neuroscientific inspiration to solve the long-sequence memory problem, thereby potentially offering insights into neuroscience observations in return.

D.2. Experiment Details

In this section, we include all experiment details including the grid search ranges and the best performing parameters. All code for reproducing the results can be found at the following repository: https://github.com/Anon-NeurIPS-2023/Wave-RNN. The code was based on the original code base from the coRNN paper (T. Konstantin Rusch and Mishra, 2021a) found at https://github.com/tk-rusch/coRNN.

Pseudocode. — Below we include an example implementation of the wRNN cell in Pytorch (Paszke et al., 2019):

```
import torch.nn as nn

class wRNN_Cell(nn.Module):
    def __init__(self, n_in, n, c, k=3):
        super(RNN_Cell, self).__init__()
        self.n = n
        self.c = c
        self.V = nn.Linear(n_in, n * c)
```

```
        self.U = nn.Conv1d(c, c, k, 1, k//2,
                           padding_mode='circular')
        self.act = nn.ReLU()

        # Sparse identity initialization for V
        nn.init.zeros_(self.V.weight)
        nn.init.zeros_(self.V.bias)
        with torch.no_grad():
            w = self.V.weight.view(c, n, n_in)
            w[:, 0] = 1.0

        # Shift initialization for U
        wts = torch.zeros(c, c, k)
        nn.init.dirac_(wts)
        wts = torch.roll(wts, 1, -1)
        with torch.no_grad():
            self.U.weight.copy_(wts)

    def forward(self, x, hy):
        hy = self.act(self.Wx(x).view(-1, self.c,
                                      self.n)
                      + self.Wy(hy))
        return hy
```

Figure 2. — The results displayed in Figure 2 are from the best performing models of the sMNIST experiments, precisely the same as those reported in Figure 7.6 (left) and Table 7.2. The hyperparamters of these models are described in the **sMNIST** section below. To compute the 2D Fourier transform, we follow the procedure of Davis et al. (2021) (Davis, G. B. Benigno, et al., 2021): we compute the real valued magnitude of the 2-dimensional Fourier transform of the hidden state activations over time (using `torch.fft.fft2(seq).abs()`). To account for the significant autocorrelation in the data, we normalize the output by the power spectrum of the spatially and temporally shuffled sequence of activations. We note that although this makes the diagonal bands more prominent, they are still clearly visible in the un-normalized spectrum. Finally, we plot the logarithm of the power for clarity.

Copy Task. — We construct each batch for the copy task as follows:

```
def get_copy_task_batch(bsz, T):
    X = np.zeros((bsz, T+10))
    data = np.random.randint(low=1, high=9,
                             size=(bsz, 10))
    X[:, :10] = data
    X[:, -(10+1)] = 9
    Y = np.zeros((bsz, T+10))
    Y[:, -10:] = X[:, :10]

    X = F.one_hot(torch.tensor(
                    X, dtype=torch.int64).permute(1,0), 10)
    Y = torch.tensor(Y).int().permute(1,0)

    return X, Y
```

For the copy task, we train each model with a batch size of 128 for 60,000 batches. The models are trained with cross entropy loss, and optimized with the Adam optimizer. We grid search over the following hyperparameters for each model type:

- iRNN
  - Gradient Clip Magnitude: [0, 1.0, 10.0]
  - **U** Initialization: [**I**, $\mathcal{U}(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}})$]
  - Learning Rate: [0.01, 0.001, 0.0001]
  - Activation: [ReLU, Tanh]
- wRNN
  - Gradient Clip Magnitude: [0, 1.0, 10.0]
  - Learning Rate [0.01, 0.001, 0.0001]

We find the following hyperparameters then resulted in the lowest test MSE for each model:

The total training time for these sweeps was roughly 1,900 GPU hours, with models being trained on individual NVIDIA 1080Ti GPUs. The iRNN models took between

Model	Parameter	Sequence Length (T)				
		0	10	30	80	480
wRNN (n=100, c=6, k=3)	Learning Rate	1e-3	1e-3	1e-3	1e-3	1e-4
	Gradient Magnitude Clip	1	0	0	1	1
iRNN (n=100)	Learning Rate	1e-3	1e-3	1e-4	1e-4	1e-4
	Gradient Magnitude Clip	10	1	1	1	10
	U-initialization	$\mathcal{U}$	**I**	**I**	**I**	**I**
	Activation	ReLU	ReLU	ReLU	ReLU	ReLU
iRNN (n=625)	Learning Rate	1e-3	1e-4	1e-4	1e-4	1e-4
	Gradient Magnitude Clip	1	10	1	1	1
	U-initialization	$\mathcal{U}$	**I**	**I**	**I**	**I**
	Activation	ReLU	ReLU	ReLU	ReLU	ReLU

Table D.1: Best performing hyperparamters for each model on the Copy Task. Gradient clipping 0 means not applied, **U** initialization $\mathcal{U}$ means Kaming Uniform Initialization.

1 to 10 hours to train depending on T and n, while the wRNN models took between 2 to 15 hours to train.

Adding Task. — For the adding task we again train the model with batch sizes of 128 for 60,000 batches using the Adam optimizer. For this task models are trained with a mean squared error loss. For both the iRNN and wRNN, we then grid-search over the following hyperparameters:

- Gradient Clip Magnitude: $[0, 1, 10, 100, 1000]$
- Learning Rate: $[0.01, 0.001, 0.0001]$

In Table D.2 we report the best performing parameters for each task setting.

The total training time for these sweeps was roughly 1,200 GPU hours. The iRNN models took between 1 to 7 hours, while the wRNN models took 6 to 12 hours each.

sMNIST. — For the sequential MNIST task we use iRNNs and wRNNs with 256 hidden units. To have a similar number of parameters, we use 16 channels with the wRNN. The models are trained with a batch size of 128 for 120 epochs. The learning rate schedule is defined such that the learning rate is divided by a factor of lr_drop_rate every lr_drop_epoch epochs. We grid search over the following hyperparameters for each model type. We find that the iRNN does not need gradient clipping on this task and achieves smooth loss curves without it. We highight in

Model	Parameter	Sequence Length (T)				
		100	200	400	700	1000
wRNN (n=100, c=27, k=3)	Learning Rate	1e-3	1e-4	1e-4	1e-4	1e-4
	Gradient Magnitude Clip	100	100	1	100	10
iRNN (n=100)	Learning Rate	1e-3	1e-3	1e-3	1e-4	1e-3
	Gradient Magnitude Clip	1000	100	10	100	1

Table D.2: Best performing settings on the Adding Task. Gradient clipping 0 means not applied. We note that the iRNN failed to solve the task meaningfully for lengths T = 700 & 1000, thus the hyperparameters found here are not significantly better than any other combination for those settings.

grey the hyperparameters which achieve the maximal performance, and were thus reported in the main text:

- iRNN
  - Learning Rate: [0.001, 0.0001, 0.00001]
  - lr_drop_rate: [3.33, 10.0]
  - lr_drop_epoch: [40, 100]
- wRNN
  - Gradient Clip Magnitude: [0, 1, 10, 100]
  - Learning Rate: [0.001, 0.0001, 0.00001]
  - lr_drop_rate: [3.33, 10.0]
  - lr_drop_epoch: [40, 100]

The total training time for these sweeps was roughly 1000 GPU hours, with models being trained on individual NVIDIA 1080Ti GPUs, iRNN models taking roughly 12 hours each, and wRNN models taking roughly 18 hours each.

psMNIST. — For the permuted sequential MNIST task, we use the same architecture and training setup as for the sMNIST task. We find that on this task the iRNN requires gradient clipping to perform well and thus include it in the search as follows:

- iRNN
  - Gradient Clip Magnitude: [0, 1, 10, 100, 1000]

  - Learning Rate: [0.001, 0.0001, 0.00001]
  - lr_drop_rate: [3.33, 10.0]
  - lr_drop_epoch: [40, 100]

- wRNN
  - Gradient Clip Magnitude: [0, 1, 10, 100, 1000]
  - Learning Rate: [0.001, 0.0001, 0.00001]
  - lr_drop_rate: [3.33, 10.0]
  - lr_drop_epoch: [40, 100]

In an effort to improve the baseline iRNN model performance, we performed additional hyperparameter searching. Specifically, we tested with larger batch sizes (120, 320, 512), different numbers of hidden units (64, 144, 256, 529, 1024), additional learning rates (1e-6, 5e-6), a larger number of epochs (250), and more complex learning rate schedules (exponential, cosine, one-cycle, and reduction on validation plateau). Ultimately we found the parameters highlighted above to achieve the best performance, with the only improvement coming from training for 250 instead of 120 epochs. Regardless, in Figure 7.6 we see the iRNN performance is still significantly below the wRNN performance, strengthening the confidence in our result. The total compute time for these sweeps was roughly 1,900 GPU hours with models being trained on individual NVIDIA 1080 Ti GPUs. The iRNN models took rougly 12 hours each, with wRNN models taking roughly 18 hours each.

For each of the wRNN models in Figure 7.7, the same hyperparameters are used as highlighted above and found to perform well. The wRNN is tested over combinations of $n = (16, 36, 64, 144) \times c = (1, 4, 16, 32)$, and the best performing models for each parameter count range are displayed. For the iRNN, we sweep over larger batch sizes (128, 512), learning rates (1e-3, 1e-4, 1e-5) and gradient clipping magnitudes (0, 1, 10, 100) in order to stabalize training, displaying the best models.

nsCIFAR10. — For the noisy sequential CIFAR10 task, models were trained with a batch size of 256 for 250 epochs with the Adam optimizer, lr_drop_epoch = 100, and lr_drop_rate = 10. We found gradient clipping was not necessary for the wRNN model on this task, and thus perform a grid search as follows, with the best performing settings highlighted:

- iRNN

  - Learning Rate: $[0.001, 0.0001, 0.00001]$
  - Number hidden units (n): $[144, 256, 529, 1024]$

- wRNN
  - Learning Rate: $[0.001, 0.0001, 0.00001]$
  - Number hidden units (n): $[144, 256]$

The total compute time for these sweeps was roughly 1,600 GPU hours with models being trained on individual NVIDIA 1080 Ti GPUs. The iRNN models took roughly 15 hours each, with wRNN models taking roughly 22 hours each.

Ablation. — For the ablation results in Table 7.4, we first report the test MSE of the best performing wRNN and iRNN models from the original Copy Task grid search (identical to those in Figure 7.3). We then added the ablation settings to the grid search, and trained the models identically to those reported above in the **Copy Task** section. This resulted in the final complete grid search:

- iRNN
  - Gradient Clip Magnitude: [0, 1.0, 10.0]
  - $\mathbf{U}$ Initialization: $[\mathbf{I}, \mathcal{U}(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}), \Sigma]$
  - $\mathbf{V}$ Initialization: $[\mathcal{N}(0, 0.001)$, Sparse-Identity$]$
  - Learning Rate: $[0.01, 0.001, 0.0001]$
  - Activation: [ReLU, Tanh]

- wRNN
  - Gradient Clip Magnitude: $[0, 1.0, 10.0]$
  - $\mathbf{u}$ Initialization: $[$Dirac, $\mathcal{U}(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}), \mathbf{u}$-shift$]$
  - $\mathbf{V}$ Initialization: $[\mathcal{N}(0, 0.001)$, Sparse-Identity$]$
  - Learning Rate $[0.01, 0.001, 0.0001]$

where Sparse-Identity refers to the $\mathbf{V}$ initialization described in section 7.2. In the case of the iRNN, the $\mathbf{V}$ matrix is defined to have 1 channel for the purpose of this initialization. The Dirac initialization is equivalent to an identity initialization for a convolutional layer and is implemented using the Pytorch function:

torch.nn.init.dirac_. We report the best performing models from this search in Table 7.4.

For the wRNN (-**u**-shift-init), we note that for the sequence lengths T where the wRNN model does solve the task (MSE $\leq 1 \times 10^{-10}$), i.e. T=0 & 10, the best performing models always use the **u** initialization $\mathcal{N}(0, 0.001)$ and appear to learn to exhibit traveling waves in their hidden state (as depicted in Figure 7.9), while the dirac initialization always performs worse and does not exhibit waves.

D.3. Additional Results

Performance Means & Standard Deviations. — In the main text, in order to make a fair comparison with prior work, we follow standard practice and present the test performance of each model with the corresponding best validation performance. In addition however, we find it beneficial to report the distributional properties of the model performance after multiple random initalizations. In Table D.3 we include the means and standard deviations of performance from 3 reruns of each of the models.

Task	Metric	iRNN	wRNN
Adding T=100	MSE	$1.40\times10^{-5} \pm 4.21\times10^{-6}$	$6.12\times10^{-5} \pm 7.95\times10^{-5}$
	Solved iter.	11,500.00 ± 2,910.33	233.33 ± 57.74
Adding T=200	MSE	$5.13\times10^{-5} \pm 1.66\times10^{-5}$	$8.54\times10^{-5} \pm 7.91\times10^{-5}$
	Solved iter.	21,000.00 ± 4,582.58	1,000.00 ± 0.00
Adding T=400	MSE	$7.70\times10^{-2} \pm 8.49\times10^{-2}$	$1.59\times10^{-4} \pm 1.07\times10^{-4}$
	Solved iter.	30,000.00 ± -	1,333.33 ± 577.35
Adding T=700	MSE	$0.163 \pm 2.08\times10^{-3}$	$5.29\times10^{-5} \pm 3.19\times10^{-5}$
	Solved iter.	×	3,000.00 ± 0.00
Adding T=1000	MSE	0.160 ± -	$4.36\times10^{-5} \pm 1.91\times10^{-5}$
	Solved iter.	×	1,666.67 ± 577.35
sMNIST	Test Acc.	98.20 ± 0.32	97.30 ± 0.34
psMNIST	Test Acc.	90.85 ± 1.47	96.60 ± 0.10
nsCIFAR10	Test Acc.	51.80 ± 0.54	54.70 ± 0.42

Table D.3: Mean and standard deviation of model performance over 3 random initializations for the best performing models in each category. We see model performance is consistent with the best performing models reported in the main text. The × means the models never solved the task after 60,000 iterations, and (± -) means that the other 2/3 random initalizations also did not solve the task after 60,000 iterations or crashed.

Additional Baseline: Linear Layer on MNIST. — One intuitive explanation for the performance of the wRNN is that the hidden state 'wave-field' acts like a register or 'tape' where the input is essentially copied by the encoder, and then subsequently processed simultaneously by the decoder a the end of the sequence. To investigate how similar the wRNN is to such a solution, we experiment with training a single linear layer on flattened MNIST images, equivalent to what the decoder of the wRNN would process if this 'tape' hypothesis were correct. In Figure D.1 we plot the results of this experiment (again showing the best model from a grid search over learning rates and learning rate schedules), and we see that the fully connected layer achieves a maximum performance of 92.4% accuracy compared with the 97.6% accuracy of the wRNN model.

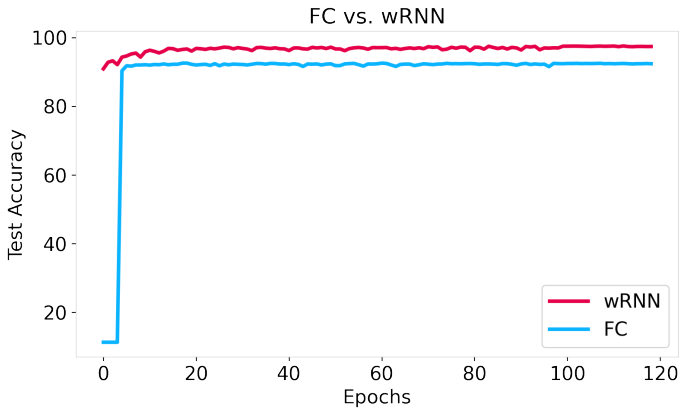


Figure D.1: Training curves for a fully connected layer trained on flattened MNIST images (FC) versus the standard wRNN presented in the main text. We see the wRNN still performs significantly better, however the FC model does mimic the rapid learning capability of the wRNN, suggesting that the wRNN's learning speed may be partially attributable to the wave-field's memory-tape-like quality.

Additional Ablation: Frozen Encoder & Recurrent Weights. — To further investigate the difference between the wRNN model and a model which simply copies input to a register, we propose to study the relative importance of the encoder weights $\mathbf{V}$ and recurrent connections $\mathbf{U}$ for the wRNN. We hypothesized that the wRNN may preserve greater information in its hidden state by default, and thus may not need to learn a flexible encoding, or perform recurrent processing. To test this, we froze the encoder and recurrent connections, leaving only the decoder (from hidden state to class label) to be trained. In Figure D.2 we plot the training curves for a wRNN and iRNN with frozen $\mathbf{U}$ and $\mathbf{V}$. We see that the wRNN performs remarkably better then the iRNN in this setting (89.0% vs. 44.3%), indicating that the wave dynamics

do indeed preserve input information by default far better than standard (identity) recurrent connections.

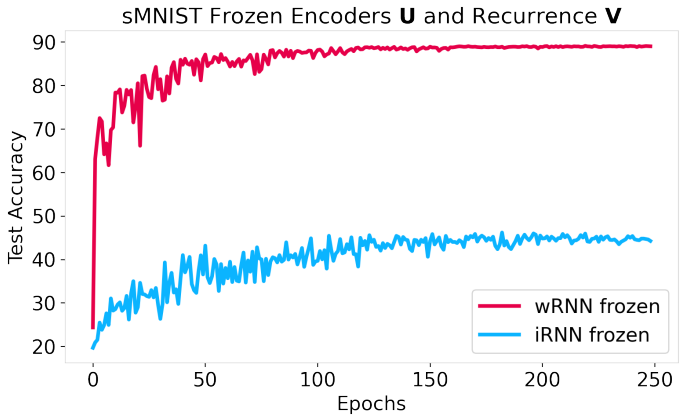


Figure D.2: Training curves for wRNN and iRNN models with frozen encoder and recurrent connections (**U** & **V**) on the sMNIST task. We see that the wRNN performs drastically better, indicating that the wRNN requires significantly less flexibility in its encoder and recurrent connections in order to achieve high accuracy.

Additional Ablation: Locally Connected RNN. — In order to test if the emergence and maintenance of waves requires the weight sharing of the convolution operation, or is simply due to local connectivity, we perform an additional experiment where we use the exact same model as the default wRNN, however we remove weight sharing across locations of the hidden state. This amounts to replacing the convolutional layer with an untied 'locally connected' layer. In practice, when initialized with same the shift-initialization we find that such a model does indeed exhibit traveling waves in its hidden state as depicted in Figure D.3. Although we notice that the locally connected network does train to comparable accuracy with the convoluitional wRNN on sMNIST, we present this preliminary result as a simple ablation study and leave further tuning of the performance of this model on sequence tasks for future work.

Additional Visualizations of Copy Task. — Here we include additional visualizations of the larger iRNN (n=625) for the copy task. We see that while it performs slightly better than the smaller iRNN (n=100) it still performs very poorly in comparison with the wRNN.

On the Emergence of Traveling Waves. — In this section we expand on the results of Figure 7.9 and include additional results pertaining to the emergence of traveling waves in recurrent neural networks with different connectivity and initialization schemes. Specifically, in Figure D.6 we show the hidden state visualization for models with varying initalizations. We see that models with shift-initialization

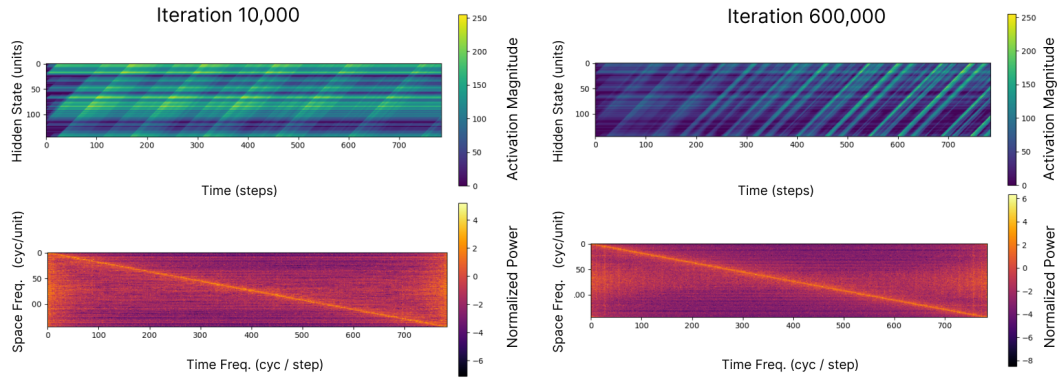


Figure D.3: Hidden state and 2D Fourier transform for a locally connected RNN showing the existence of traveling waves. These results imply that it is simply local connectivity which is important for the emergence of traveling waves rather than shared weights.

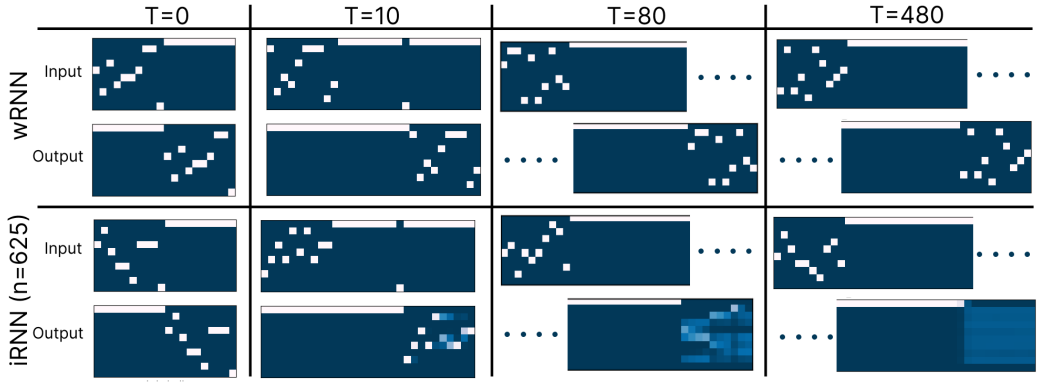


Figure D.4: Examples from the copy task for the larger iRNN (n=625) and wRNN (n=100, c=6) as shown in the main text. We see the larger iRNN does slightly better than the small iRNN (n=100) shown in Figure 7.4, but still significantly worse than the wRNN. These results clearly show that the iRNN does not have the appropriate machinery for storing memories over long sequences while the wRNN does.

exhibit waves directly from initialization, while randomly initialized convolutional models do not initially exhibit waves but learn to exhibit them during training, and identity initialized models never learn to exhibit waves. Furthermore, in Figure D.5 we show the respective training curves for randomly initialized and identity initialized wRNNs. We see that the randomly initialized wRNN achieves higher final accuracy in correspondence with the emergence of traveling waves, reinforcing the conclusion that traveling waves are ultimately beneficial for sequence modeling performance.

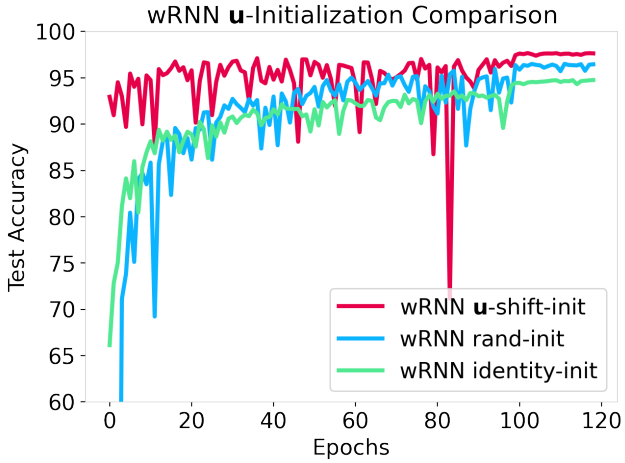


Figure D.5: Training curve on the sMNIST task for a wRNN with three different initialization schemes: **u**-shift (default), random ($\mathcal{U}(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}})$), and identity (dirac). We see that the random initalization does not have the same rapid learning speed as the **u**-shift initialization, however, it does still achieve significantly higher final accuracy than the identity initialization, implying traveling waves are beneficial to performance.

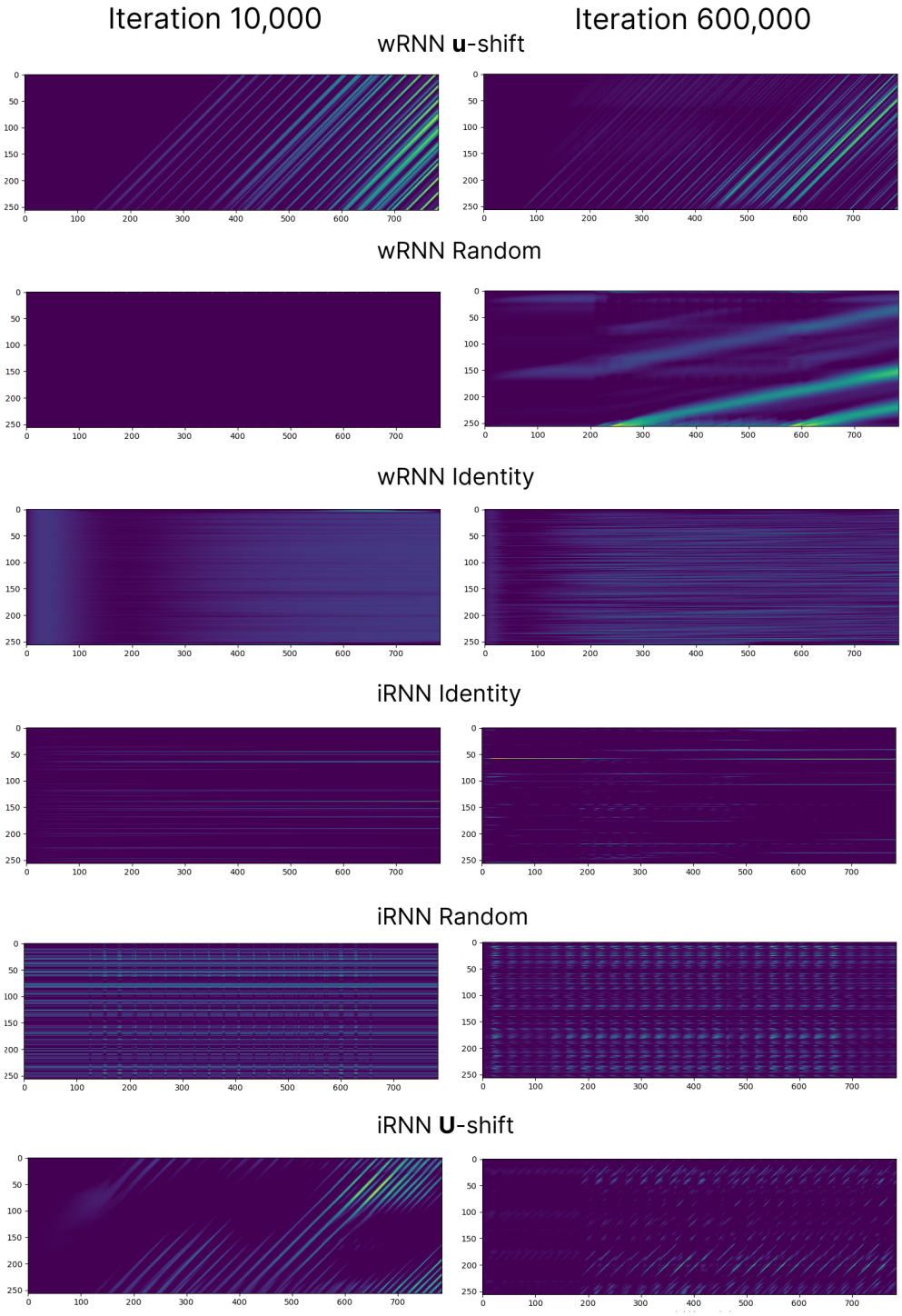


Figure D.6: Visualization of hidden state (y-axis is hidden units) over timesteps (x-axis) for a variety of different models and initalizations for **U**. We see that the wRNN with **u**-shift initialization achieves the most consistent waves throughout training. Interestingly, some other models learn to achieve traveling waves despite not having them at initalization (wRNN with random (kaming uniform) initialization); while other models, (iRNN with **U**-shift) initially have stronger traveling waves, and slowly lose them throughout training. We see the wRNN with identity (dirac) initialization never learns waves despite using convolutional recurrent connections.

Appendix E

CHAPTER VIII APPENDIX

E.1. Implementation details

Common settings. — During the training stage, we randomly sample one single transformation at each iteration. The batch size is set to 128 for both datasets. We use Adam optimizer and the learning rate is set as $1e{-}4$ for all the parameters. The encoder consists of four stacked convolution layers with the activation function ReLU, while the decoder is comprised of four stacked transposed convolution layers. For the prior evolution, the diffusion coefficient D_k is initialized with 0 and we set it as a learnable parameter for distinct k. For `MLPs` that parameterize the potential $u(\boldsymbol{z}, t)$ and the force $f(\boldsymbol{z}, t)$, we use the sinusoidal positional embeddings (Vaswani et al., 2017) to embed the timestep t, and use linear layers for embedding the latent code $\boldsymbol{z}$. `Tanh` gates are applied as the activation functions of the `MLPs`. All the experiments are run on a single NVIDIA Quadro RTX 6000 GPU.

MNIST. — The input images are of the size 28×28. The sequence of each transformation contains 9 states of variations. The scaling transformation scales the image from 1.0 up to 1.8 times. The rotation transformation rotates the object by maximally 80 degrees, and the coloring transformation adjusts the image hue from 0 to 340 degrees. The model is trained for $90,000$ iterations.

Shapes3D. — The input images are resized to 64×64. Each transformation sequence consists of 8 images. The model is also trained for $90,000$ iterations.

Weakly-supervised setting. — For the `Gumbel-Softmax` trick, we re-parameterize $q_\gamma(k|\bar{\boldsymbol{x}})$ by

$$y_i = \frac{e^{\frac{x_i+g_i}{\tau}}}{\sum_i e^{\frac{x_i+g_i}{\tau}}} \tag{E.1}$$

where x_i is the category prediction, g_i is the sample drawn from Gumbel distributions, and τ is the small temperature to make `softmax` behave like `argmax`. We take the 'hard' binary prediction in the forward pass and use the straight-through gradient estimator (Bengio, Léonard, et al., 2013) during backpropagation. The temperature τ is initialized with 1 and is gradually reduced to 0.05 with the annealing rate $3e{-}5$.

Baselines. — For the disentanglement methods, we largely enrich the original MNIST dataset by adding the transformed images of the whole sequence. This makes it possible for both β-VAE and FactorVAE to learn the given transformations in an unsupervised manner. For tuning the interpolation range, we start from the initial value z_i and traverse till the appropriate bound which is selected from the range $[-5, 5]$ with the interval of 0.1.

Disentanglement metrics. — There are some traditional disentanglement metrics (Ridgeway and Mozer, 2018; Eastwood and Williams, 2018; R. T. Chen et al., 2018), but they are designed for single-dimension traversal methods. When it comes to vector-based disentanglement methods such as (Shen and Zhou, 2021; Tzelepis et al., 2021; Y. Song, T. A. Keller, et al., 2023), the scores would drop considerably and cannot be compared with those single-dimension baselines. Therefore, we directly evaluate our method using the equivariance error instead of disentanglement metrics.

E.2. Ablation studies

We conduct two ablations to study the impact of different priors and PDE constraints.

Table E.1: Equivariance error of different priors.

Prior	Scaling	Rotation	Coloring
SG	190.24±2.18	158.93±3.25	164.18±2.77
MoG	188.23±2.45	157.79±2.86	161.49±2.62
VAMP	192.81±3.67	161.47±4.12	162.97±3.89
Diffusion	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**

Table E.2: Equivariance error of different PDEs.

Prior	Scaling	Rotation	Coloring
Heat	223.95±3.38	212.47±3.85	207.66±2.91
FP	211.54±3.17	188.59±3.92	194.73±3.09
OHJ	190.43±2.48	163.87±3.03	162.38±2.86
GHJ	**185.42±2.35**	**153.54±3.10**	**158.57±2.95**

Impact of different priors. — We use diffusion equations to model the prior evolution as random particle movement. It would also be intriguing to choose other priors commonly used in the VAE literature, such as Standard Gaussian (SG) priors $\mathcal{N}(0, 1)$, mixture of Gaussian (MoG) priors $\sum w_i \mathcal{N}(\mu_i, \sigma_i^2)$, and VAMP priors (Tomczak and Max Welling, 2018) which average aggregated posterior of N pseudo-inputs as $1/N \sum_n q(\boldsymbol{z}_n)$. Table E.1 presents the equivariance error of different priors on MNIST. Among these priors, our diffusion equations achieve the best performance. This meets our assumption that modeling the prior evolution as a diffusion process suits more the random motion. Nonetheless, we see that the performance gap between each baseline is narrow, which somehow implies that the impact of different priors is limited.

Impact of different PDEs. — We apply the generalized HJ (GHJ) equation as the PINN constraint in order to achieve dynamic OT. It would be also interesting to try other commonly used PDEs. We compare our GHJ with the ordinary HJ (OHJ) equation, the Fokker Planck (FP) equation, and the heat equation. Table E.2 compares the equivariance error of PDEs on MNIST. Our GHJ and OHJ equations achieve the best performance as they both satisfy the condition of dynamic OT. This empirical evidence indicates that the OT theory can indeed model better latent flow paths. Moreover, our GHJ outperforms the OHJ by a slight margin. We attribute this advantage to the external driving force $f(\boldsymbol{z}, t)$ which gives us more flexibility and dynamics in modeling the velocity fields ∇u^k.

E.3. HJ equations as dynamic optimal transport

We now turn to introduce why HJ equations could minimize the Wasserstein distance. As stated in (Benamou and Brenier, 2000), the L_2 Wasserstein distance can be re-formulated in the fluid mechanical interpretation as

$$W^2 = \inf \int_D \int_0^1 \frac{1}{2} \rho(x,t) v(x,t)^2 \, dx \, dt \tag{E.2}$$

where the density satisfies the continuity equation ($\partial_t \rho = -\nabla \cdot (\rho(x,t)v(x,t))$. If we introduce the momentum $m(x,t) = \rho(x,t)v(x,t)$ and two Lagrange multipliers u and λ, the Lagrangian function of the Wasserstein distance would be:

$$L(\rho, m, \phi) = \int_D \int_0^1 \frac{||m||^2}{2\rho} + u(\partial_t \rho + \nabla \cdot m) - \lambda(\rho - s^2) \tag{E.3}$$

where the second term is the equality constraint, the third term is the inequality constraint ($\rho > 0$), and s is a slack variable. Using integration by parts formula, the above equation can be re-written as

$$L(\rho, m, \phi) = \int_D \int_0^1 \frac{||m||^2}{2\rho} + \int_D u\rho|_0^1 - \int_D \int_0^1 (\partial_t u\rho + \nabla u \cdot m) - \lambda(\rho - s^2) \tag{E.4}$$

Based on the set of Karush–Kuhn–Tucker (KKT) conditions ($\partial_m L = 0$, $\partial_u L = 0$, $\partial_\rho L = 0$, and $\lambda \geq 0$), we would have:

$$\begin{cases} \partial_m L = \frac{m}{\rho} - \nabla u = v - \nabla u = 0 \\ \partial_u L = \partial_t \rho + \nabla \cdot m = 0 \\ \partial_\rho L = -\frac{||m||^2}{2\rho^2} - \partial_t u - \lambda = -\frac{1}{2}||v||^2 - \partial_t u - \lambda = 0 \end{cases} \tag{E.5}$$

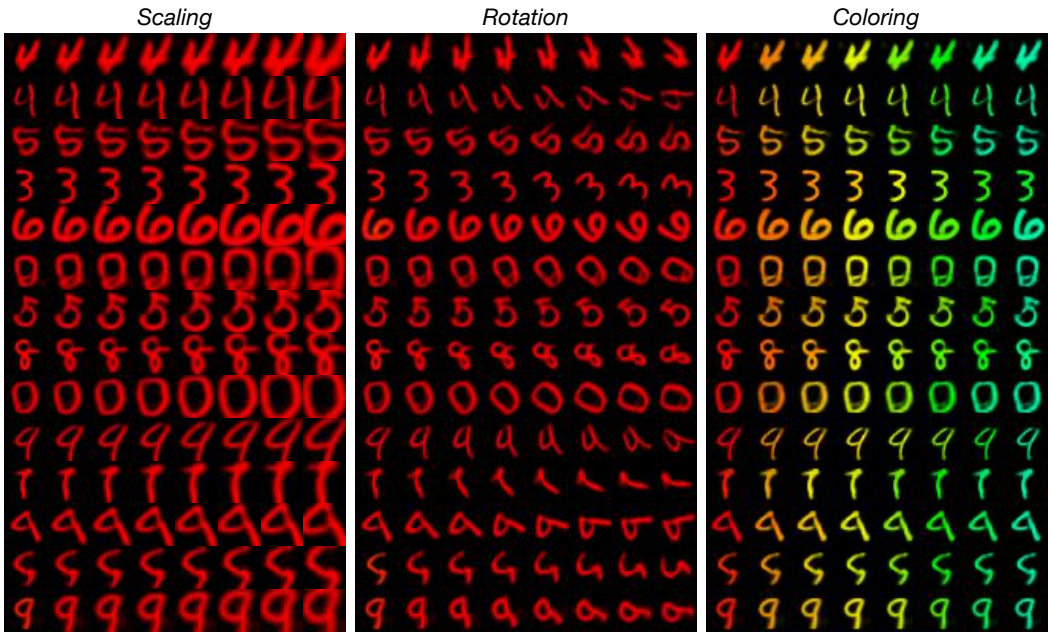


Figure E.1: More visualizations of the learned latent flows on MNIST (LeCun, 1998).

where the first condition indicates that the gradient ∇u acts as the velocity field, and the third condition implies the optimal solution is given by the generalized HJ equation:

$$\partial_t u + \frac{1}{2}||\nabla u||^2 = -\lambda \leq 0 \tag{E.6}$$

We thus apply the generalized HJ equation (*i.e.,* $\partial_t u + \frac{1}{2}||\nabla u||^2 \leq 0$) as the constraints. We further use an extra negative force because this would give more dynamics for modeling the posterior flow.

E.4. More visualizations

Fig. E.1 and E.2 display more visualization results of the latent evolution on MNIST and Shapes3D, respectively. On both datasets, our method presents precise control of the given transformations. Fig. E.3 and E.4 show more latent evolution results of switching transformations (top) and combining transformations (bottom) on MNIST and Shapes3D, respectively. Our latent flows learn to compose or switch different transformations precisely and flexibly.

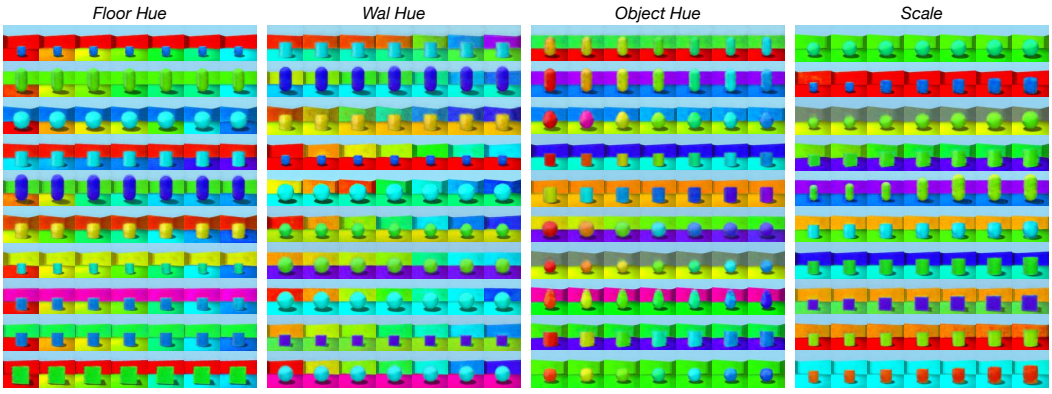


Figure E.2: More visualizations of the learned latent flows on Shapes3D (Burgess and H. Kim, 2018).

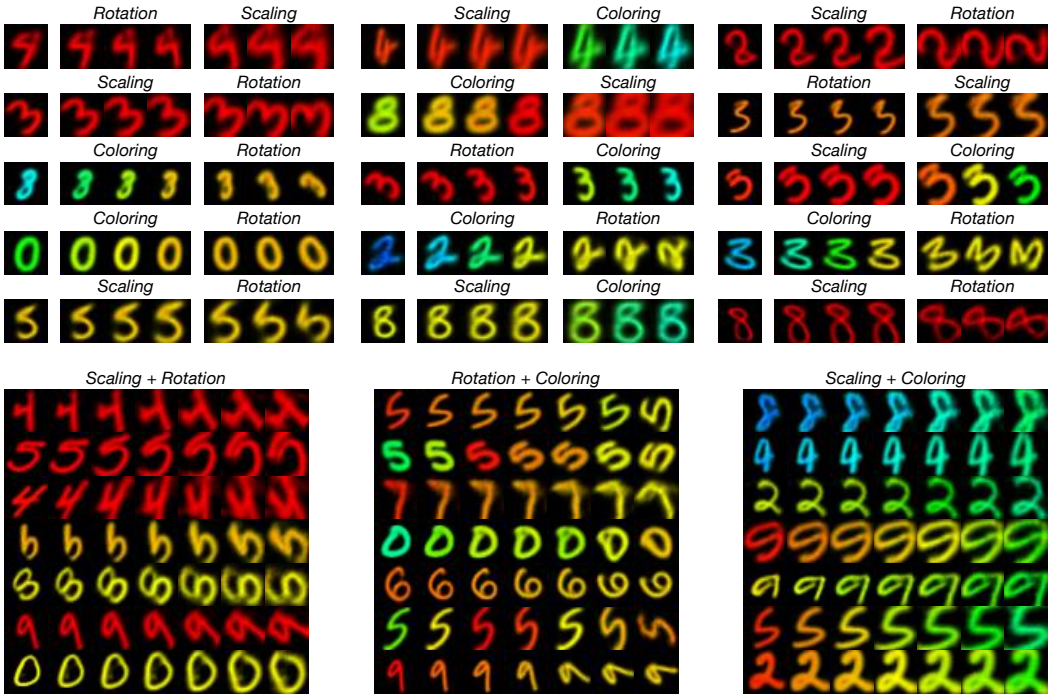


Figure E.3: More visualizations of switching and superposing transformations on MNIST (LeCun, 1998).

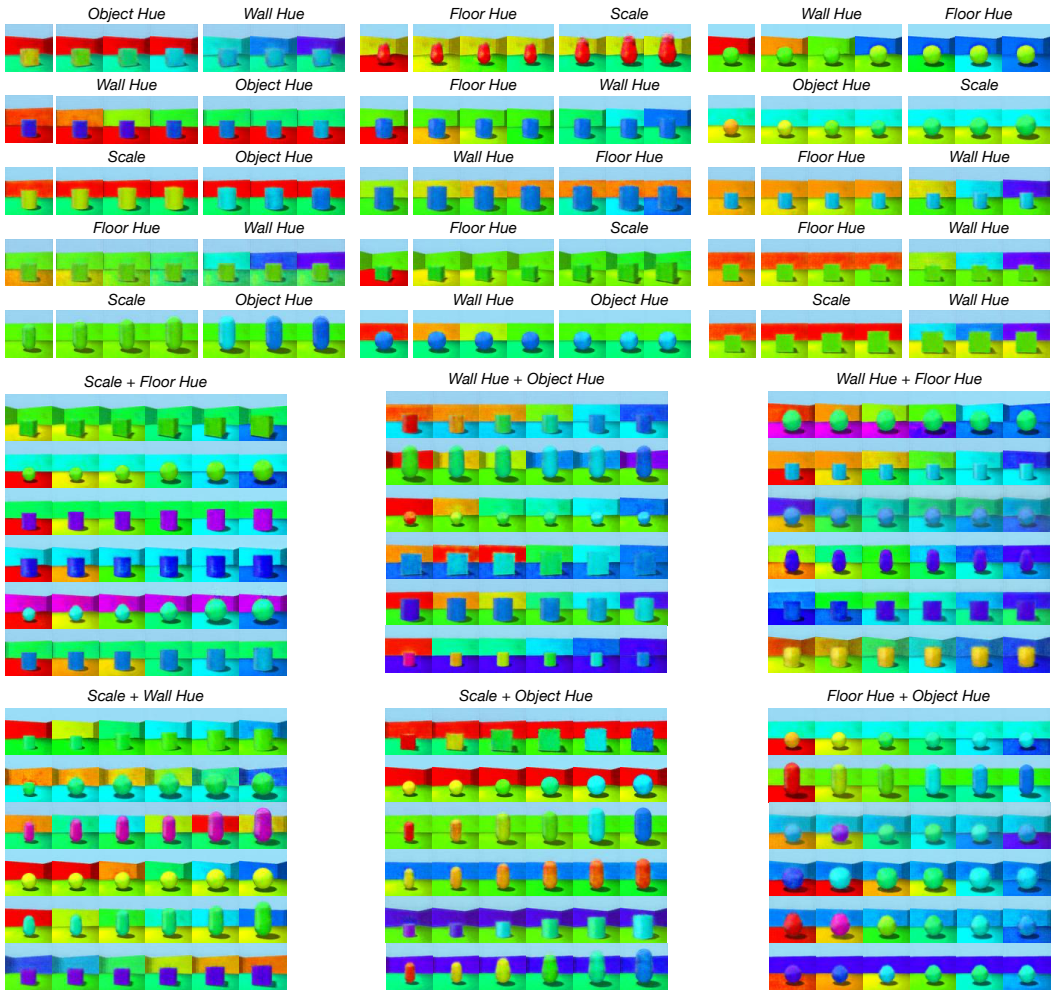


Figure E.4: More visualizations of switching and superposing transformations on Shapes3D (Burgess and H. Kim, 2018).

Appendix F

CHAPTER IX APPENDIX

F.1. Experiment Details

Model Architectures — All models presented in this paper were built using the convolutional layers from the SESN (Sosnovik et al., 2020) library for consistency and comparability (`https://github.com/ISosnovik/sesn`). For scale equivariant models, we used the set of 6 scales $[1.0, 1.25, 1.33, 1.5, 1.66, 1.75]$. To construct the rotation equivariant backbones, we use only a single scale of $[1.0]$ and augment the basis set with four 90-degree rotated copies of the basis functions at $[0^o, 90^o, 180^o, 270^o]$. These rotated copies thus defined the group dimension. This technique of basis or filter-augmentation for implementing equivariance is known from prior work and has been shown to be equivalent to other methods of constructing group-equivariant neural networks (B. Li et al., 2021). For translation models, we perform no basis-augmentation, and again define the set of scales used in the basis to a single scale $[1.0]$, thereby leaving only the spatial coordinates of the final feature maps to define the output group.

On MNIST (LeCun and Cortes, 2010), we used a backbone network f composed of three SESN convolutional layers with # channels (32, 64, 128), kernel sizes (11, 7, 7), effective sizes (11, 3, 3), strides (1, 2, 2), padding (5, 3, 3), no biases, basis type 'A', BatchNorm layers after each convolution, and ReLU activations after each BatchNorm. The output of this final ReLU was then considered our z for contrastive learning (with $\mathcal{L}_{A-SSL}$ and $\mathcal{L}_{H-SSL}$) and was of shape $(128, S \times R, 8, 8)$ where S was the number of scales for the experiment (either 1 or 6), and R was the number of rotation angles (either 1 or 4). For experiments where the transformation studied was not translation, we average pool over the spatial dimensions before applying the projection head h to achieve a consistent dimensionality of 128. For classification, an additional SESN convolutional layer was placed on top with kernel size 7, effective size 3, stride 2, and no padding, thereby reducing the spatial dimensions to 1, and the total dimensionality of the input to the final linear classifier to 128.

On CIFAR10 we used a ResNet20 model composed of an initial SESN lifting layer with kernel size 7, effective size 7, stride 1, padding 3, no bias, basis type 'A', and 9 output channels. This lifted representation was then processed by a following

SESN convolutional layer of kernel size 7, effective size 3, stride 1, padding 3, no bias, basis type 'A', and 64 output channels. This initial layer was followed by a BatchNorm and ReLU before being processed by three ResNet blocks of output sizes (128, 256, 512) and initial strides of (1, 2, 2). Each ResNet block is composed of 3 SESN Basic blocks as defined here (`https://github.com/ISosnovik/sesn/blob/master/models/stl_ses.py#L19`). The output of the third ResNet block was taken as our $\boldsymbol{z}$ for contrastive learning (again for $\mathcal{L}_{A-SSL}$ and $\mathcal{L}_{H-SSL}$) of shape $(512, S \times R, 7, 7)$. Again, as for MNIST, for experiments where the transformation studied was not translation, we average pool over the spatial dimensions before applying the projection head h to achieve a consistent dimensionality of 512. For classification, the vector $\boldsymbol{z}$ was first max-pooled along the scale/rotation group-axis ($S \times R$), followed by a BatchNorm, a ReLU, and average pooling over the remaining 7×7 spatial dimensions. Finally, we apply BatchNorm to this 512-dimensional vector before applying the non-linear projection head h.

On Tiny ImageNet we use a Resnet20 model which has virtually the same structure as the CIFAR10 model, but instead uses 4 ResNet blocks of output sizes (64, 128, 256, 512) and strides (1, 2, 2, 2). Furthermore, each ResNet block is composed of only 2 BasicBlocks for TIN instead of 3 for CIFAR10. Overall this results in a $\boldsymbol{z}$ of shape $(512, S \times R, 4, 4)$, and a final vector for classification of size 512. We note that we do not include Translation results in Table 9.1 for Tiny ImageNet precisely because the spatial dimensions of the feature map with this architecture are too small to allow for effective H-SSL training in the settings we used for other methods.

All models used a detached linear classifier for computing the reported downstream classification accuracies, while the Supervised baselines used an attached linear layer (implying gradients with respect to the classification loss back-propagated though the whole network). All models additionally used an attached non-linear projection head h constructed as an MLP with three linear layers. For MNIST these layers have of output sizes (128, 128, 128), while for CIFAR10 and TIN they have sizes (512, 2048, 512). There is a BatchNorm after each layer, and ReLU activations between the middle layers (not at the last layer).

Training Details — For training we use the LARS optimizer with an initial learning rate of 0.1, and a batch size of 4096 for all models. We use an NCE temperature (τ) of 0.1, half-precision training, a learning rate warm-up of 10 epochs, a cosine lr-update schedule, and weight decay of 1×10^{-4}. On MNIST we train for 500 epochs and on CIFAR10 and Tiny ImagNet (TIN) we train for 1300 epochs. On

average each MNIST run took 1 hour to complete distributed across 8 GPUs, and each CIFAR10/TIN run took 10 hours to complete distributed across 64 GPUs. In total this amounts to roughly 85,000 GPU hours.

Empirical Validation — For the experiments in Table 9.1, we use two different methods for data augmentation, and similarly two different methods for selecting the representations ultimately fed to the contrastive loss for the A-SSL and H-SSL settings.

For A-SSL we augment the input at the pixel level by: randomly translating the image by up to ± 20% of its height/width (for translation), randomly rotating the image by one of $(0^o, 90^o, 180^o, 270^o)$ (for rotation), or randomly downscaling the image between 0.57 and 1.0 of its original scale. For S-SSL we use no input augmentations.

For both methods we use only a single fiber, meaning the base size $|\boldsymbol{g}_0|$ is 1. For A-SSL, we randomly select the location $\boldsymbol{g}_0$ for each example, but we use the same $\boldsymbol{g}_0$ between both branches. For example, in translation, we compare the feature vectors for two translated images *at the same pixel location*. Similarly, for scale and rotation, we pick a single scale or rotation element to compare for both branches. For H-SSL, we randomly select the location $\boldsymbol{g}$ independently for each example *and independently for each branch*, effectively mimicing the latent operator.

H-SSL Without Structure — In Table 9.2, we use the same overall model architectures defined above (3-layer model or ResNet20), but replace the individual layers with non-equivariant counterparts. Specifically, for the MLP, we replace the convolutional layers with fully connected layers with outputs (784, 1024, 2048). For the convolutional models (denoted CNN $(6 \times CHW)$), we replace the SESN kernels with fully-parameterized, non-equivariant counterparts, otherwise keeping the output dimensionality the same (resulting in the $6\times$ larger output dimension).

Furthermore, for these un-structured representations, in the H-SSL setting, we 'emulate' a group dimension to sample 'fibers' from. Specifically, for the MLP we simply reshape the 2048 dimensional output to (16,128), and select one of the 16 rows at each iterations. For the CNN, we similarly use the 6 times larger feature space to sample $\frac{1}{6}^{th}$ of the elements as if they were scale-equivariant.

Parameters of H-SSL — For Figure 9.3 (left), we select patches of sizes from 1 to $|G| - 1$ with no padding. In each setting, we similarly increase the dimensionality of the input layer for the non-linear projection head h to match the multiplicative increase in the dimension of the input representation $\boldsymbol{z}(\boldsymbol{g})$. For the topographic

distance experiments (right), we keep a fixed base size of $|\boldsymbol{g}_0| = 1$ and instead vary the maximum allowed distance between randomly sampled pairs g_1 & g_2.

F.2. Related Work

Our work is undoubtedly built upon the the large literature base from the fields equivariant deep learning and self-supervised learning as outlined in Sections **??** and 9.2. Beyond this background, our work is highly related in motivation to a number of studies specifically related to equivariance in self-supervised learning. Most prior work, however, has focused on the undesired invariances learned by A-SSL methods (Xiao et al., 2021; Tsai, T. Li, et al., 2022) and on developing methods by which to avoid this through learned approximate equivariance (Dangovski et al., 2022; Wang, Geng, et al., 2021). Our work is, to the best of our knowledge, the first to suggest and validate that the primary reason for the success of feature-space SSL objectives such as DIM(L) (Hjelm et al., 2019) and GIM (Löwe, O'Connor, et al., 2019) is due to their exploitation of equivariant backbones.

F.3. Broader Impact

This work is primarily related to understanding and improving self-supervised learning – a training method for deep neural networks which is able to leverage large amounts of unlabeled data from the internet, making it one of the most used methods for state of the art image and text generative models today (Radford, J. W. Kim, et al., 2021; Ramesh, Pavlov, et al., 2021). Such models have significant broader impact and potential negative consequences which are beyond the scope of this work. We refer readers to discussions of those paper for further information. Specifically, this work aims to improve such SSL techniques, thereby inheriting the broader impact of these models.