

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

A Group Theoretic Framework for Neural Computation

Permalink

<https://escholarship.org/uc/item/6fj3m1tk>

Author

Sanborn, Sophia

Publication Date

2021

Peer reviewed|Thesis/dissertation

A Group Theoretic Framework for Neural Computation

by

Sophia Sanborn

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Psychology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Steve Piantadosi, Chair

Professor Bruno Olshausen

Professor David Whitney

Dr. Christopher Hillar

Fall 2021

A Group Theoretic Framework for Neural Computation

Copyright 2021

by

Sophia Sanborn

Abstract

A Group Theoretic Framework for Neural Computation

by

Sophia Sanborn

in Psychology

University of California, Berkeley

Assistant Professor Steve Piantadosi, Chair

How do networks of neurons compute stable representations of continually transforming sense data? This thesis aims to provide a mathematical foundation for this question based in group theory—the mathematics that describes much of the transformation structure in natural visual signals. The primary contribution of this thesis is a framework that ties group theory to neurophysiologically plausible computational mechanisms by way of generalized Fourier analysis and its roots in group representation theory. I demonstrate mathematically and computationally how *group representations* may be instantiated in biological neurons to achieve equivariance to signal transforms. I further demonstrate how complete, selective group invariance can be achieved in a third-order generalization of the classical energy model of complex cells, based on the bispectrum. This perspective offers a reframing and generalization of canonical models of receptive fields in visual cortex, explains extra-classical results, and makes novel empirical predictions. I additionally demonstrate the utility of this mathematical framework for building artificial neural networks with improved robustness and invariance and apply these architectures to the problems of learning latent transformation structure from data and group-invariant denoising of signals in associative memory networks.

Contents

Contents	i
1 Introduction	1
1.1 The Problem: How We Know Universals	3
1.1.1 Power Spectral Approaches	5
1.1.2 Max & Average Pooling Approaches	6
1.1.3 Factoring	8
1.1.4 Disentangling	11
1.2 Group Theory and Perception	12
1.3 Overview	14
2 Mathematical Foundations	16
2.1 Groups	17
2.2 Isomorphisms and Homomorphisms	18
2.3 Commutativity	19
2.4 Lie Groups	20
2.5 Matrix Groups	21
2.6 Group Actions	22
2.7 Homogeneous Spaces	23
2.8 Invariance and Equivariance	23
2.9 Representations	23
2.10 Harmonic Analysis	25
3 A Group Theoretic Framework for Neural Computation	28
3.1 The Problem: Invariance and Completeness	29
3.2 A Universal Solution: The Bispectrum	30
3.3 Proposal: Group Representations as Universals	35
3.4 Evidence in Biology	36
3.4.1 Equivariance: Group Representations in Visual Cortex	36
3.4.2 Invariance: Bispectral Hyper-Complex Cells	42
3.5 Discussion	46

4	Learning Groups and Their Representations in Neural Networks	47
4.1	The Problem: Learning a Complete Invariant Map	48
4.2	Model: Bispectral Neural Networks	48
4.3	Experiment 1: Supervised Learning of the Bispectrum	49
4.3.1	Methods	51
4.3.2	Results	53
4.3.3	Discussion	56
4.4	Experiment 2: Learning Group Invariant Maps	57
4.4.1	Methods	59
4.4.2	Cyclic Translation $S1$	60
4.4.3	2D Rotation $SO(2)$	71
4.4.4	3D Rotation $SO(3)$	80
4.5	Summary and Conclusions	84
5	Group Invariant Associative Memory	85
5.1	The Problem: Group Invariant Memory and Denoising	85
5.2	Model: Bispectral Hopfield Networks	87
5.3	Experiment 1: Memory Capacity for Random Phasor Patterns and Their Translations	89
5.3.1	Methods	89
5.3.2	Results	90
5.3.3	Discussion	92
5.4	Experiment 2: Translation Invariant Denoising of Phase Structure in Natural Images	92
5.4.1	Methods	92
5.4.2	Results	93
5.4.3	Discussion	94
5.5	Summary and Conclusions	94
6	Discussion	96
	Bibliography	98

Acknowledgments

I started my Ph.D. interested in characterizing the notion of *generalization* or *structural similarity* at a cognitive level, inspired by the work of psychologists such as Roger Shepard. Indeed, my graduate school admissions essays to cognitive psychology programs proposed making progress on the topic of *analogical reasoning*. I'm ending my Ph.D. now with a thesis that ties the general concepts underlying these ideas to the mathematics of groups, their possible neural implementation in the visual system by way of group representations, and their utility for building machine learning systems for robust perception. This journey has taken me around and through many topics in mathematics, vision science, neuroscience, signal processing, computer science, and machine learning—ultimately landing back where I started, new tools in hand.

This journey would not have been possible without the pairing of freedom and support that I have received at Berkeley. Coming in with an NSF Graduate Research Fellowship allowed me to pursue my interests in the first three years without the constraints of grant funding. My early advisors in the psychology department—Alison Gopnik and Tom Griffiths—were patient and supportive as I followed my interests wherever they led me. The psychology department graciously allowed me to fulfill many of my degree requirements with courses across campus.

It was through these courses that I was fortunate to encounter Bruno Olshausen and the Redwood Center for Theoretical Neuroscience, where I came upon the level and style of explanation I was after. Bruno's rare combination of big-picture vision with low-level technical rigor is something to aspire to. I am incredibly grateful to have had him as a mentor and role model as I developed my own scientific identity. The many long afternoons we've spent in scientific discussion have been a major highlight of my graduate career, and I look forward to many more.

My fortune continued upon meeting Chris Hillar, maverick mathematician, who first introduced me to the concept of the bispectrum and has been a primary collaborator on much of my work since. Chris has not only been a constant source of inspiration, but has also been incredibly generous with his time and mentorship throughout my studies, providing immense support in my mathematical and scientific training. His impact on this work and my development as a researcher cannot be overstated.

Along the way, I have had many excellent collaborators who have made the research process a pleasure. I am especially grateful to have had Dylan Paiton as my first collaborator at Redwood, to whom I am indebted for his thoughtful patience as he mentored my first project in computational neuroscience, and Christian Shewmake

as my newest collaborator, who has been an absolute joy to collaborate with on both technical work and long-term vision.

I was also fortunate to contribute to industrial research during internships at Intel during my Ph.D, first in the AI group and then in the Neuromorphic Computing Lab. Thank you to Amir Khosrowshahi, Paxon Frady, Garrick Orchard, and Mike Davies for the opportunity to work at the intersection of scientific research and cutting edge industrial technology.

Throughout graduate school, I've been lucky to be embedded in a community that has infused my daily life with art and inspiration. I am most grateful to: My partner Filip Victor, whose authenticity, growth mindset, and belief in the power of curiosity inspires me every day—for the many years of growing and learning together, for all of the pressing philosophical debates and books that belong in the freezer, for meandering nor cal hikes, global adventures, and future visions.

To Mariel Goddu, grad school sister and true friend, for the early days mind mapping ideas on large rolls of brown paper, to long Bolinas beach walks and psychological investigations, to Tuesdays in the Berkeley hills.

And to my community at Agape, for creating the feeling of home, and for providing countless opportunities for spontaneous artistic collaboration: To the kitchen couch and living room piano. To Power 105, the Studio Collective, and AV Club for persistently pulling me away from research for jam sessions, synth experiments, algoraves, renegade shows, and immersive art. Thank you for all of the beautiful moments of creation.

And lastly, I am incredibly grateful to my brother Ben Sanborn and my mother Cheryl Sanborn, for many things, but especially for the deep love of nature that was instilled in me by our family culture.

Thank you all for coming along this journey with me.

Chapter 1

Introduction

To them, I said, the truth would be literally nothing but the shadows of the images.

Plato
Allegory of the Cave, Republic, 375 B.C.

A remarkable property of biological vision is that organisms maintain stable recognition of objects that project highly variable patterns of light on the retina at each instant. During vision, objects and patterns continually undergo transformations such as translations, rotations, scaling, deformations, and changes in lighting. Despite these variations, organisms effortlessly and robustly recognize objects and parse complex scenes, demonstrating *perceptual invariance*.

Moreover, organisms are able to *explicitly model* transformations in a scene—for example, extrapolating trajectories and recognizing novel poses of objects. This is despite the fact that sensory signals provide only partial, highly underdetermined glimpses into this transformation structure. Visual signals, critically, arise from the lossy 2-dimensional projection of objects transforming in the inherently 3-dimensional space of the physical world. Like the prisoners in Plato’s *Allegory of the Cave*, our visual systems must infer the structure of reality from mere ‘shadows on the wall.’ What are the mathematics underlying computations in networks of neurons as they accomplish this feat?

This thesis approaches this question with the mathematics that describes transformation structure in the natural world: group theory, and the adjacent fields of representation theory and harmonic analysis. In the natural world, variations are highly

structured, arising from the symmetry and geometry of the space in which objects lie. Consequently, many of the transformations in natural data can be described in terms of the actions of *groups*. The premise of this thesis is that the computations performed by neural circuits reflect this group structure—the *mathematical symmetries* of the natural world. Group theory was instrumental in unifying models of the fundamental forces and elementary particles in 20th-century physics. Likewise, it has the potential to unify our understanding of how neural circuits model the world.

The relevance of this mathematics for perception has been noted sporadically throughout 20th century—early on by Henri Poincaré [101], formalized into an early neural network model by Pitts and McCulloch [100], and later appreciated by William Hoffman [49] and Roger Shepard [117] as a framework for understanding perceptual phenomena. The work of these theorists inspired some empirical studies in psychology [13] and neuroscience [38] as well as several neural network models [107, 85, 21, 120]. However, by and large, the group theoretic approach has failed to gain traction as a mainstream framework for understanding neural computation.

One likely reason for this is the relative inaccessibility of the underlying mathematics for researchers in psychology and neuroscience, as group theory and related topics (topology, differential geometry, harmonic analysis, and representation theory) are not typically studied outside of a mathematics or physics curriculum. A second, related cause is that software libraries dedicated to computing with the objects of group theory are few and far between. Consequently, a researcher aiming to make use of this mathematics for stimulus generation, data analysis, or computational modeling must implement most of it from scratch.

Primarily, however, the field has lacked a framework of concrete, neurophysiologically plausible, empirically testable mechanisms for how neural circuits might implement the mathematics of group theory. Hoffman’s noteworthy Lie group framework for vision [49] provides a largely high-level mathematical description of perceptual phenomena such as perceptual illusions.¹ Likewise, Shepard used group theory to pose perceptual experiments [117], but did not posit underlying neurophysiological mechanisms. The computational models of [107, 85, 21, 120]—while important computational advances—are not intended as biological models, and all involve neurophysiologically implausible computations.

The primary contribution of this thesis is the advancement of a framework that ties the

¹In later work, Hoffman proposes an interesting theory of neuroanatomical structure in the visual cortex based in Lie theory and differential geometry [50]. However, it makes few predictions in terms of neural computation.

mathematics of group theory to neurophysiologically plausible computational mechanisms by way of generalized Fourier analysis and its roots in group representation theory. My goal here is to demonstrate mathematically and computationally how *group representations* may be instantiated in biological neurons to achieve robust invariance and equivariance to signal transforms. I argue that group representations provide universal solutions to problems in perception and demonstrate how this framework generalizes canonical models of receptive fields in visual cortex, explains extra-classical results, and makes novel empirical predictions (Chapter 3). I then demonstrate the utility of this mathematical framework for building artificial neural networks with improved robustness and invariance (Chapters 4 and 5).

1.1 The Problem: How We Know Universals

An object is displaced before my eye, and its image which was first formed on the centre of the retina is now brought to the edge of the retina . . . These sensations are conducted to me by two different nerves. They ought to appear to me different in character . . . Why, then, do I come to conclude that the *same* image has been displaced?

Henri Poincaré
On the Foundations of Geometry, 1898

How do we recognize objects when they appear with different scales, positions, orientations, lighting, deformations, and distortions? This is the problem of *perceptual invariance*. At the heart of the challenge lie two requirements that are frequently in tension: *invariance* and *selectivity*. Our visual systems are invariant in highly specific ways. Neural representations selectively encode structures that are relevant for behavior, while factoring out *all* and *only all* transformations on that structure that are behaviorally irrelevant.

Consider the problem of recognizing the face of a friend. Changes in lighting, orientation, distance, and position of the face in the visual field are all irrelevant for recognition and can be disregarded for this task. Even more complex transformations, such as those resulting from emotion or aging can be disregarded. However, any alterations to the precise structural pattern that defines facial identity will alter the recognition response. Just how this is accomplished by the primate visual system is still an open question, despite decades of research.

The canonical, though incomplete, picture of how this is achieved in visual cortex is that of hierarchical feature extraction and pooling. The idea traces back to the

work of Hubel and Wiesel in the 1960s on simple and complex cells (Figure 1.1) [56, 57, 55]. Their landmark studies identified neurons (*simple cells*) in the cat primary visual cortex that can be reliably stimulated by bars oriented in particular directions. These neurons are highly selective for the width, orientation, and position of the bar. Continuous changes to these parameters continually decrease and then eliminate the cells' responses. Another class of neurons (*complex cells*) exhibit simultaneous *invariance* to the position of the bar and *selectivity* for its width and orientation. Hubel and Wiesel hypothesized that complex cells may work by combining the outputs of two simple cells whose preferred positional encoding is shifted.

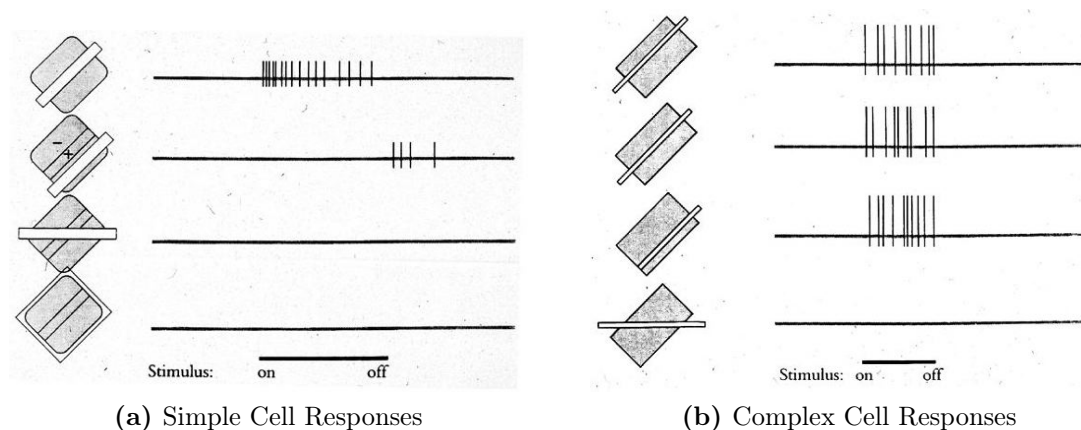


Figure 1.1: Simple and Complex Cells from the studies of Hubel and Wiesel [55]. Since the original studies, the basic phenomena have been replicated many times, most notably [88, 30, 130, 129, 112].

This idea set the foundation for what has become the canonical model of neural computation in the ventral stream. The essence of the model is this: through successive stages of computation in the the visual cortex, neurons extract features that are progressively more complex and more invariant, by combining the outputs of neurons in the stages below [55, 36]. Over several decades, empirical researchers have accumulated a substantial body of data that appear to corroborate this perspective, including examples of neurons in higher visual areas (e.g. V4 and IT) that selectively code for faces, hands, and objects, while exhibiting invariance to position, lighting, size, and other visual transformations [43, 109, 105].

Two classes of models have had particular importance in establishing the computational side of this theory. The first—which I call *power spectral approaches*—are analytically constructed from hand-specified wavelet features (typically Gabor or Morlet

wavelets), with invariance achieved by eliminating the phase structure of these transforms [102, 1, 11]. The second—which I call *max & average pooling approaches*—learn visual features from data in an optimization process, while achieving increasing invariance in successive layers via max or average pooling [36, 74, 108]. This latter class of models grew into the deep convolutional neural networks that now dominate the field of image recognition [72].

1.1.1 Power Spectral Approaches

The basic power spectral approach was proposed as a model of translation invariance in complex cells—first suggested conceptually by Pollen, et al. (1971) [102], and then concretely demonstrated in a computational model by Adelson & Bergen (1985) [1]. The *energy model* of complex cells is built on a Gabor model of simple cells, in which simple cell receptive fields are represented as localized, Gaussian-windowed 2D Fourier filters (Gabor wavelets) that linearly transform inputs—a classical model that reproduces much of the phenomena associated with simple cells [80]. The responses of complex cells are then computed as the sum of the squared magnitude of the output of a pair of simple cells whose receptive fields are in quadrature phase (i.e., the real and imaginary components of a complex Gabor filter). This is equivalent to computing a localized Fourier power spectrum. Figure 1.2 depicts the approach.

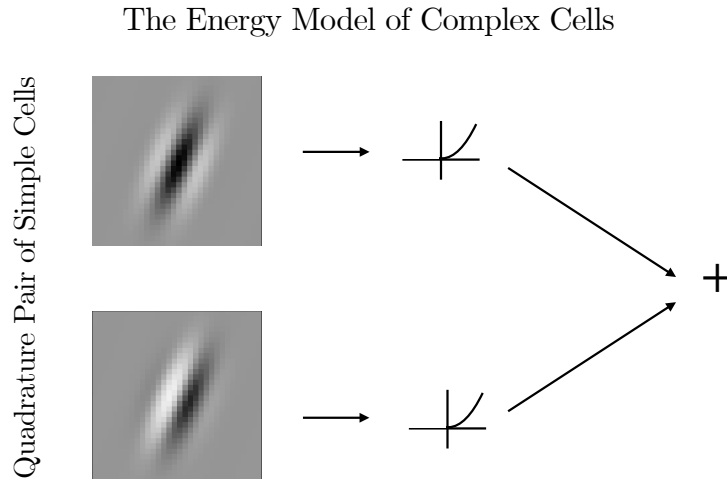


Figure 1.2: The Energy Model of Complex Cells. A single complex cell is modeled as the half-squared sum of the outputs of two quadrature phase simple cells.

A key issue with this model is that the power spectrum is a highly *non-selective* invariant. To achieve invariance in the power spectrum, the *phase* of the Fourier

coefficients is discarded. This eliminates any phase shifts due to the translation of a signal in pixel space (i.e. due to the Fourier shift property), but it also eliminates all information about the phase structure in a signal. Consequently, “adversarial” examples can be generated that have equivalent power spectra but very different image structure (Figure 1.3). If the goal is robust, selective invariance, this is highly undesirable, as the majority of meaningful structure is contained in a signal’s relative phase structure [96]. Indeed, there is neurophysiological data to suggest that complex cells do not simply eliminate phase structure [31, 46, 83, 58, 138]—a point that we will examine more thoroughly in Chapter 3. Despite this, the energy model has remained the classical model of translation invariant complex cells.

Though arising from a different body of literature in signal processing, the general approach to constructing invariant signal representations has been extended to a hierarchical context in the Scattering Wavelet Transform [11]. By interleaving stacked wavelet transforms and “modulus pooling” (preserving the average magnitude and eliminating the phase of the complex coefficients), the model generates multi-scale translation-invariant representations of signals that are stable with respect to deformations in the input. While the output at any given layer is phase-insensitive, the information is passed through to higher levels, so the model can distinguish between some signals with identical power spectra. However, there is an inherent loss of information in the modulus pooling operation that prevents the transform from being fully invertible, except under specific conditions [12]. Thus, the transform still lacks selectivity.

The appeal of the power spectral approach is that it can be analytically defined and computed for translation. To extend this approach to other transformations relevant to perception, other theorists have turned to a combination of feature learning and pooling through a *max* or *avg* operation, to achieve invariance to arbitrary transformations on arbitrary features. This approach set the foundation for what has become deep learning.

1.1.2 Max & Average Pooling Approaches

The deep learning approach can be traced back to Fukushima’s 1980 *Neocognitron*, arguable the first deep convolutional neural network [36]. Inspired by Hubel and Wiesel’s simple and complex cells, the architecture of the Neocognitron consists of hierarchically stacked layers, with interleaved feature detection neurons (simple cells) and pooling neurons (complex cells) (Figure 1.4). Each simple cell extracts visual features from a local region of its inputs. The feature it computes is defined by the neuron’s synaptic weights to the input, which are learned through optimization. Complex cells are connected to a fixed set of simple cells that extract the same feature

The Power Spectrum is a Non-Selective Invariant

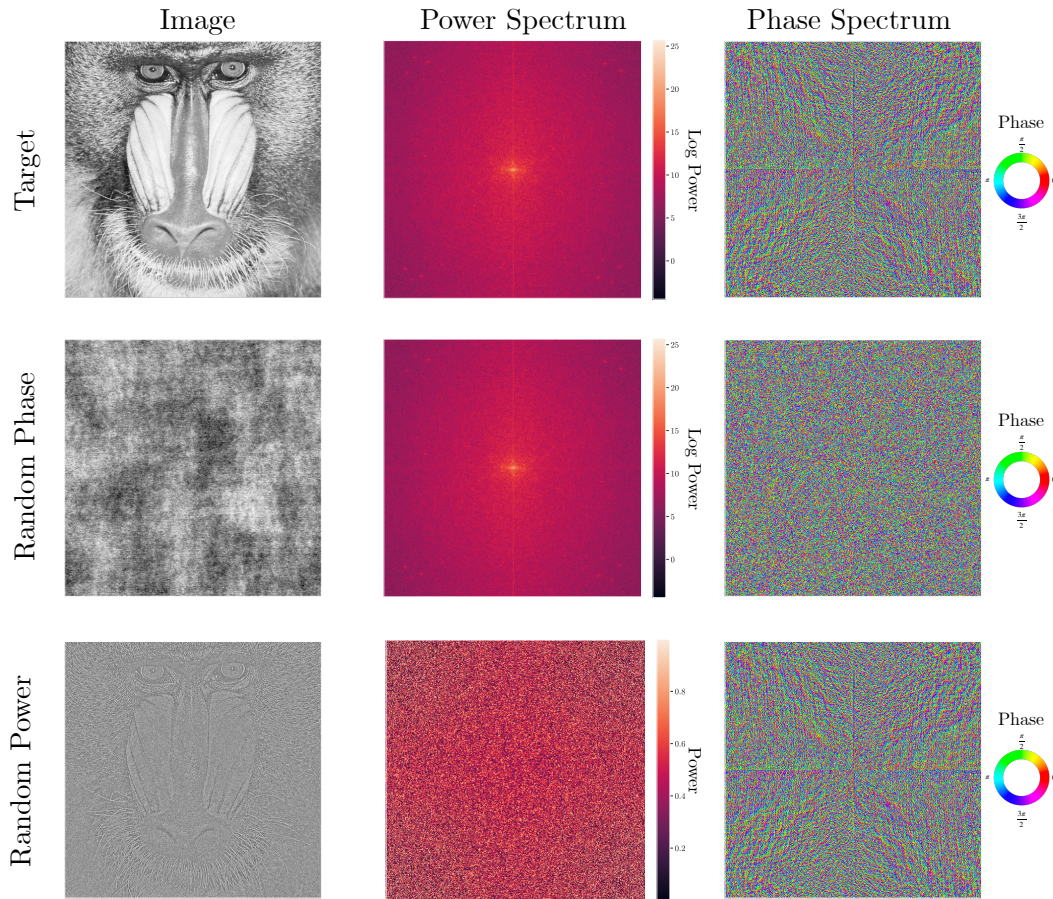


Figure 1.3: The Power Spectrum is a Non-Selective Invariant. Two images with identical power spectra (top and middle) can have very different image structure. However, images with identical phase spectra and non-identical power spectra (top and bottom) share much of the meaningful image structure—that which defines edges and contours. May require enlargement to see. Note that the sharp horizontal and vertical lines in the power and phase spectra are artifacts due to the wrap-around of the image boundaries.

in slightly shifted positions. The complex cell outputs a response if at least one of the connecting simple cells is active. Thus, it maintains a constant response as a feature is spatially translated across the region that it pools. Other, more complex perturbations to the neural outputs will also yield the same or similar responses, giving this network invariance to the broader set of transformations underlying a task such as character recognition.

Fukushima proposed several methods for training the network on this task, with the network generally learning edge-like features in the first layer and combinations of these edges in the higher layers. Modern deep neural networks are typically trained with backpropagation, but the general approach of achieving invariance by pooling neural outputs remains the same. To date, the vast majority of state-of-the-art deep networks architectures incorporate pooling layers throughout the stack, with pooling neurons taking either the maximum or average value of the outputs of a localized subset of neurons in the previous layer (Figure 1.5).

Unfortunately, these pooling methods offer no better guarantees for selectivity than the power-spectrum-based energy model. Just like the power spectrum, these pooling operations are *invariant* but not *selective*. The *max* and *avg* operations are invariant to *all* structural changes, including random permutations of the values in the layers below. The result is a system is built on highly lossy operations that do not selectively preserve perceptually relevant information. This loss of structural information likely underlies many of the well-known failure modes of deep image recognition networks, such as susceptibility to adversarial examples and perturbations. Indeed, many results indicate that even state-of-the-art systems are simultaneously too invariant and not invariant enough [24, 121, 8, 86, 60] (Figure 1.6).

Despite their incompleteness, both the power spectral and deep learning approaches have withstood the test of time as paradigmatic models of invariant visual processing. Two other, related ideas have also played important roles in understanding how the brain models transformation structure in sense data: the notions of *factoring* and *disentangling*. These approaches emerged to meet the goal of *representing* rather than *eliminating* transformation structure.

1.1.3 Factoring

The notion of separating *what* (pattern structure) from *where* (transformations) has a long history in neuroscience, including classical neurophysiological results suggesting distinct processing streams for object identity and spatial information in the ventral and dorsal streams [40]. Computational theorists have posed the problem as *factoring*—decomposing information into elementary patterns and transformations that are

The Neocognitron

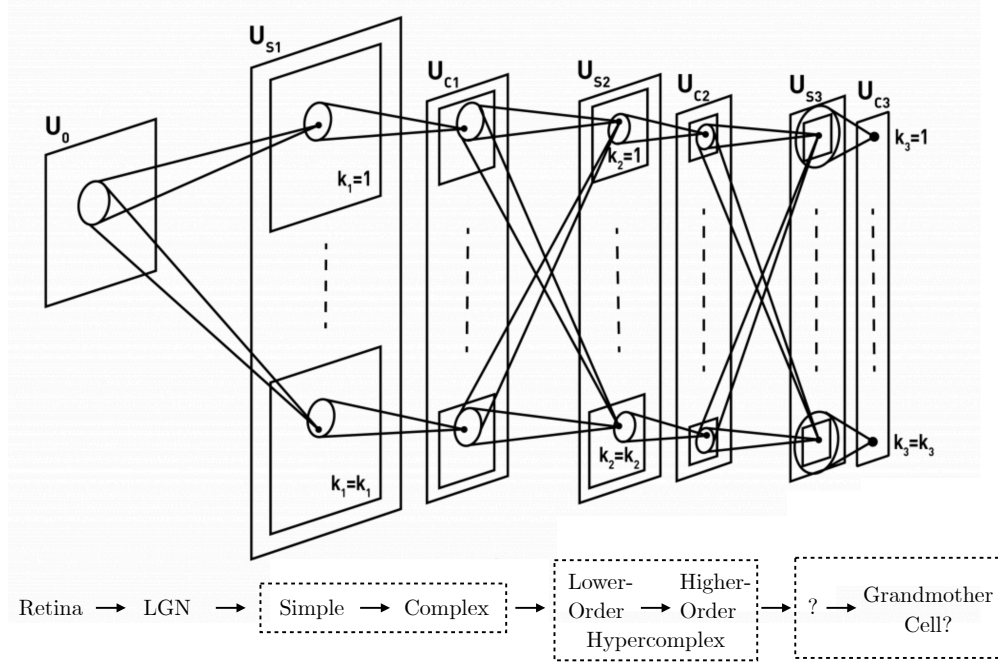


Figure 1.4: The Neocognitron. Features are extracted and pooled at successively higher stages of processing, mirroring the simple, complex, and hypercomplex stages proposed by Hubel and Wiesel [57]. Groups of S neurons (analogous to simple cells) and C neurons (analogous to complex cells) work together in hierarchical stages to extract local features (S) and pool the responses of groups of neurons with identical features in shifted positions (C). Adapted from Figures 1 and 2 in [36].

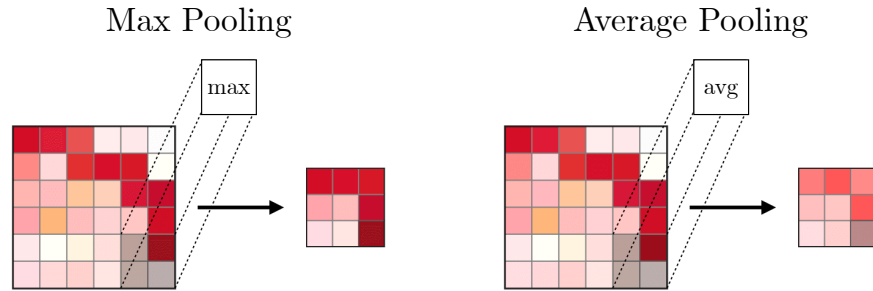


Figure 1.5: Max and Average Pooling. Pooling neurons connect to a local region of the outputs of the previous layer, summarizing the region by taking either the maximum (left) or average value (right). This reduces the dimensionality of the output and is invariant to perturbations in the input. Diagrams from the Stanford CS 230 Convolutional Neural Networks Cheatsheet.

Excessive Invariance in Deep Neural Networks



Figure 1.6: Excessive Invariance in Deep Neural Networks. Pairs of images in the top and bottom rows produce identical logit activations in the output of a high-performing trained ResNet, revealing a lack of perceptually relevant selectivity in the learned neural representations. Figure adapted from Jacobsen, et al. (2018) [60].

multiplicatively combined to generate the observed stimuli. A primary advantage of the *factoring* approach is that transformations are *explicitly modeled*. This has benefits for generalization: a transformation can be combined with a pattern to recognize transformed versions of it after observing only one exemplar. Factoring approaches are thus typically generative models in which data are represented with and generated from the multiplicative combination of latent variables.

One notable approach is *dynamic routing* [48, 2, 93, 3]. In dynamic routing models, a set of “control neurons” correct for transformations in the input by rerouting the information using multiplicative gates to higher levels of processing, thus placing the signal into a canonical reference frame. Such architectures belong to the broader category of bilinear models or third-order Boltzmann machines [127, 42, 84, 95, 21]. These latter models have demonstrated how the multiplicative couplings necessary for factoring *content* from *style*, or *pattern* from *transformation*, can be learned in an unsupervised manner from visual input. The idea of factoring has been incorporated into modern deep learning in several recent models that aim to explicitly parameterize and learn transformations in trainable neural network modules [61, 140, 104, 23, 53, 113, 7].

Across all of these models, however, the mathematical structure necessary to define and model transformations in the input is typically not rigorously defined or accounted for. Many kinds of transformation structure are difficult or impossible to represent

in generic neural network layers. Consequently, such networks will be limited in the transformation structure they can learn or represent.

1.1.4 Disentangling

Related to, yet distinct from, the notion of *factoring* is the notion of *disentangling*. Similarly to factoring approaches, disentangling approaches aim to separate the factors of variation that generate the various exemplars of an object. Distinctly, however, disentangling approaches do not *explicitly model* transformation structure in separate modules or control structures. Rather, disentangling models aim to learn a latent space in which factors of variation are separated, typically loaded onto independent components along which continuous changes in values correlate with transformations in the input. This results in an *implicit* representation of the transformation structure present in the data, in which high-dimensional, correlated, non-linear actions in the pixel space become lower-dimensional, decorrelated, and linearized in the latent space.

Under this definition, the classical models of Principal Component Analysis [99] and Independent Component Analysis [59] can be viewed as simple disentangling approaches. Manifold learning approaches such as IsoMap [126], LLE [110], and UMAP [82] also fall under this category. Many deep learning approaches for disentangling have also been proposed, typically using decorrelation constraints built into the loss function, as in InfoGAN [16] and Beta-VAE [47], among others [15, 76, 77, 66].

In visual neuroscience, the concept of *disentangling object manifolds* [116, 27] has served as a motivating framework for understanding invariant object recognition, using a slightly different notion of *disentangling*. The basic premise is that the set of all transformations on an individual visual object forms an inherently low-dimensional manifold. In the retinal image space, the manifolds of many objects are “entangled” and high-dimensional. The proposed goal of visual processing for object recognition is to transform the data into a lower-dimensional neural representation in which distinct object manifolds are disentangled and thus linearly classifiable.

This approach has motivated the use of dimensionality reduction techniques to uncover latent manifold structure in neural population responses in both biological and artificial neural networks [17]. However, to date, the idea has served largely as a high-level framework rather than as a source of mechanistic hypotheses for how computations in individual neurons give rise to the low-dimensional, continuous dynamics described as neural manifolds.

The ideas presented in this thesis are first steps towards those mechanistic hypotheses.

Here, I propose a revival of *group theory*—the mathematics that describes much of the transformation structure in natural visual signals—for understanding the neural computational mechanisms underlying invariant perception.

1.2 Group Theory and Perception

[Nervous structures] serve to classify information according to useful common characters. In vision, they detect the equivalence of apparitions related by similarity and congruence, like those of a single physical thing seen from various places. In audition, they recognize timbre and chord, regardless of pitch. The equivalent apparitions in all cases share a common figure and define a **group of transformations** that take the equivalents into one another but preserve the figure invariant. . . . These figures are the *geometric objects* of Cartan and Weyl, the *Gestalten* of Wertheimer and Köhler.

Walter Pitts & Warren S. McCulloch
How We Know Universals, 1947

Theorists as far back as Poincaré have noted the relevance of group theory for the perceptual invariance problem. In the 1898 text *On the Foundations of Geometry* [101], Poincaré motivates a treatise on geometry with the perceptual invariance problem—puzzling over how it is that the brain recognizes visually transforming objects as *the same* when they result in distinct patterns of stimulation on the retina. He suggests, loosely, that the brain may compensate for displacements in visual input with an internal (neural) displacement that counters the transformation, thus canonicalizing the input in the internal representation. Critically, Poincaré’s poses the problem in terms of the actions of *groups*.

In the 1940s—against the backdrop of the early 20th-century successes of group theory in physics—Pitts and McCulloch also tackled the perceptual invariance problem, proposing the first neural computational model of perceptual invariance [100] shortly after introducing the first artificial neural network [81]. Like Poincaré, they based their analysis in the mathematics of groups.

At a high level, a group is a set of transformations that meet certain algebraic conditions.² Many of the geometric transformations relevant for perception such as translation, rotation, and scaling can be described as groups—*Lie groups*, in particular,

²We will review the mathematics of groups in detail in Chapter 2.

which are groups of differentiable transformations that lie on a manifold. Lie groups and their associated objects—representations, generators, algebras—give us the mathematics to precisely define, represent, and parameterize the transformation structure in signals and the problem of perceptual invariance.

Unfortunately, Pitts and McCulloch’s 1947 model is not much more than the concept of average pooling couched in group theory. The basic idea is to achieve invariance to a group of transformations by averaging over all transformed versions of a signal—to *average over the group*. Just like conventional average pooling, this approach is woefully non-selective and would give the same response to signals with permuted internal structure. Taking an average is a near-trivial invariant that preserves little of the meaningful information in the signal. We shall see in Chapter 3 that *selective invariance* requires rather specific conditions. Nonetheless, the mathematical framing of the problem and attempt to tie it to neuroanatomy was commendable, especially in those very early years of neuro- and computer science.

William Hoffman was perhaps the next to pick up the thread, noting the relevance of Lie groups for mathematically defining the “basic perceptual invariances,” in his 1966 paper, *The Lie Algebra of Visual Perception* [49]. Hoffman goes on to hypothesize that many perceptual phenomena, such as orthogonal after-images [78] and the spiral after-effect [35], can be explained in terms of Lie group differential operators. The idea was picked up by perceptual psychologists Ferraro and Caelli, who conducted novel perceptual experiments to test these ideas [13, 32, 33]. Hoffman stopped short of positing neural computational mechanisms. However, his work did inspire an interesting study of receptive field properties in primary cortex [38], which we will examine in Chapter 3.

In other work, Roger Shepard emphasized the importance of transformation groups and geometry as priors for perception [117] and posited the “mental representation” of this geometric structure as an explanation for the phenomenon of “mental rotation” [118]. In studies of mental rotation, human subjects are asked to determine whether two 3D objects are equivalent up to rotation. Shepard and others found a linear relationship between the subject’s reaction time and the angular distance between equivalent rotated objects. While this suggests an important computational role for geometric transformations, Shepard did not offer any hypotheses for how this might be implemented in the brain.

In the 1990s and 2000s, however, a number of papers in computer vision sprung up making use of Lie group theory for defining [134] and solving [107, 85, 120] perceptual invariance problems. These latter models made use of the notion of the Lie algebra, aiming to learn the Lie algebra generators of a group in generative models that factor

patterns from their transformations generated through the exponential map.

One notable challenge with these approaches is the computational complexity of computing gradients through the matrix exponential. Some avoided this complexity by using a first-order Taylor expansion to approximate the exponential map [107, 85]. However, this linear approximation is accurate only for small transformations. In other work [120], the Lie algebra generator was restricted to a diagonal matrix to reduce computational complexity while computing the full exponential map (which is trivial for a diagonal matrix, as it is equivalent to the scalar exponential of each diagonal entry). This diagonalization approach was later formalized and generalized by Cohen & Welling [19], who drew connections to group representation theory and noted that the restriction explicitly defines the set of compact, connected, commutative (CCC) Lie groups, all of which have diagonalizable generators over the complex field. A later paper [14] combined this approach with sparse coding to learn irreducible representations of CCC groups.

A related line of work has emerged in the deep learning community, stemming from a demand for generalizing translational convolution—which is equivariant to the group of translations—to non-Euclidean domains such as graphs and 3D surfaces—in which the convolutional analogs are permutation and 3D rotation, respectively. This motivated the incorporation of additional group symmetries into neural architectures [18, 70, 128, 20] and has grown into its own subfield, under the name Geometric Deep Learning [10]. These approaches typically require choosing groups *a priori* and explicitly building that structure into the network architecture, to ensure equivariance to the actions of those groups.

Unfortunately, all of these approaches offer no better solution to the problem of *selective invariance*. Invariance in these models is either not addressed at all, is achieved through conventional max pooling, or is achieved through a Pitts-and-McCulloch-style lossy group averaging approach. This underscores the difficulty of this problem. Despite decades of research spanning biology, mathematics, and computer science, we still lack a computational theory for perception that captures the highly nuanced selective invariance that biological systems exhibit.

1.3 Overview

This thesis aims to provide a foundation for a new approach to understanding robust equivariance and invariance in biological and artificial neural networks through group representation theory and the generalized bispectrum. Here, I propose that *group theory*—the mathematics that describes much of the transformation structure in nat-

ural visual signals—can be used to (1) more clearly and accurately define the problem of invariant object recognition; (2) provide a set of mechanistic hypotheses for how transformation structure and invariance may be represented and achieved through computations in individual neurons; and (3) construct artificial neural networks that are simultaneously *selective* and *invariant*, faithfully respecting both pattern and transformation structure in data. The thesis is structured as follows:

In **Chapter 2**, I introduce the mathematical foundations on which this framework rests, covering relevant concepts in group theory, representation theory, and harmonic analysis.

In **Chapter 3**, I introduce the framework, precisely define the problem of selective invariance from a group theoretic perspective, and propose two hypotheses: (1) that neurons in the visual cortex compute the representations of groups relevant for perception, with classical Gabor receptive fields a special case of this more general phenomenon and (2) that neurons in the visual cortex compute the bispectral coefficients of signals to achieve robust invariance that is simultaneously selective for relative phase structure and invariant to absolute phase. I review empirical evidence that supports these hypotheses and make several novel predictions.

In **Chapter 4**, I use this mathematical framework to propose a novel neural network architecture for learning arbitrary compact groups from data and computing robust perceptual invariants. For data structured by both commutative (i.e. translation and 2D rotation) and non-commutative (i.e. 3D rotation) groups, the networks converge to canonical solutions involving approximations of group representations, and learn to approximate the bispectra of these groups. The learned solutions exhibit improved robustness and selectivity as compared to generic multi-layer neural networks trained on the same tasks.

In **Chapter 5**, I demonstrate how this framework can be applied to the problem of storing patterns in associative memory networks that can generalize one-shot to transformations of those patterns. The Bispectral Hopfield Network generalizes the learning rule of complex phasor networks [92] to mirror the third-order ansatz of the group invariant bispectrum, resulting in an associative memory network that only needs to store a single pattern to denoise all transformed versions of it. Due to the general nature of the bispectrum, this model can be constructed for arbitrary commutative and non-commutative compact groups, though I demonstrate it only for 2D translation here.

Chapter 6 makes space for discussion and conclusions.

Chapter 2

Mathematical Foundations

Before diving in, I first briefly cover the basic mathematical foundations of the framework, in group theory, representation theory, and harmonic analysis. Particularly important are the connections between groups, their representations, and generalized Fourier analysis, as reviewed in Sections 2.9 and 2.10. Readers familiar with these topics can comfortably skip ahead to Chapter 3.

Group theory is a branch of mathematics that uses algebraic methods to formalize and study the concept of *symmetry*. Simple geometric symmetries are familiar to everyone who has been through grade-school. The symmetries of a shape are the actions one can perform on it that leave its appearance invariant. We are interested in precisely characterizing the *symmetries of sensory signals*—the transformations on objects in the world that leave sensory structure invariant.

Though geometric symmetry is perhaps the most immediately intuitive, the abstract concepts of symmetry and invariance have proved useful across many areas of mathematics. Indeed, the foundations of group theory emerged in parallel out of geometry, number theory, and the theory of algebraic equations. Famous 18th and 19th century mathematicians such as Euler, Gauss, Lagrange, Cauchy, Galois, Abel, Lie, Poincaré, and Cartan were all involved in developing its foundations. However, it was not until Klein’s *Erlangen Program* proclaimed group theory as the organizing principle of geometry that it came to center stage in mathematics. Not long after, through the work of Hilbert, Weyl, and Noether—later Atiya, Yang, and Mills—20th century physics was revolutionized through the unifying principle of symmetry.

In this section, I cover the many concepts from group theory that we need to establish a mathematical formulation of perceptual invariance. Of course, we can only just begin to scratch the surface of these topics here; the purpose is simply to lay down the

requisite concepts and definitions that are used in the later theoretical developments. I make an attempt to emphasize intuition and examples over proofs here, as the basic intuitions are most important for understanding how these ideas apply to signal processing and sensory neuroscience.

2.1 Groups

Formally, a **group** $(G, *)$ is a set G with a binary operation $*$, which we can generically call the *product*. The notation $a * b$ denotes the product of two elements in the set; however, we will typically omit the operator and write simply ab . Note that the product $*$ is an abstract binary operation and does not imply any relationship to the familiar multiplicative product $\times$. The symbol $*$ is simply a placeholder, and can be replaced with any other arbitrary symbol. Indeed, we can (and will) refer to other familiar operations such as addition $+$ as an abstract product.

For our purposes, it is useful to think of the group G as defining a class of transformations, such as a finite set of two-dimensional rotations. The elements of the group $g \in G$ define *particular* transformations, such as *rotation by 30°* or *rotation by 90°* . The binary operation $*$ provides a means for combining two particular transformations—for example, first rotating by 30° and then rotating by 90° . For a set of transformations G to define a group under the operation $*$, the four following axioms must hold:

1. **Closure:** The product of any two elements of the group is also an element of the group, i.e. for all $a, b \in G$, if $ab = c$, then $c \in G$.
2. **Associativity:** The grouping of elements under the operation does not change the outcome, so long as the order of elements is preserved, i.e. $(ab)c = a(bc)$.
3. **Identity:** There exists a “do-nothing” *identity* element e such that the product of e with any group element g returns g , i.e. $ge = eg = g$ for all $g \in G$.
4. **Inverse:** For every group element g , there exists an *inverse* element g^{-1} such that the product of g and g^{-1} returns the identity, i.e. $gg^{-1} = g^{-1}g = e$.

A simple example is the set of integers under addition, commonly denoted $(\mathbb{Z}, +)$. The sum of any two integers is also an integer, so closure is satisfied. The grouping of operations makes no difference in the end result: $(a + b) + c = a + (b + c)$, i.e. it is associative. There exists an identity element 0 that returns n for any $n + 0$.

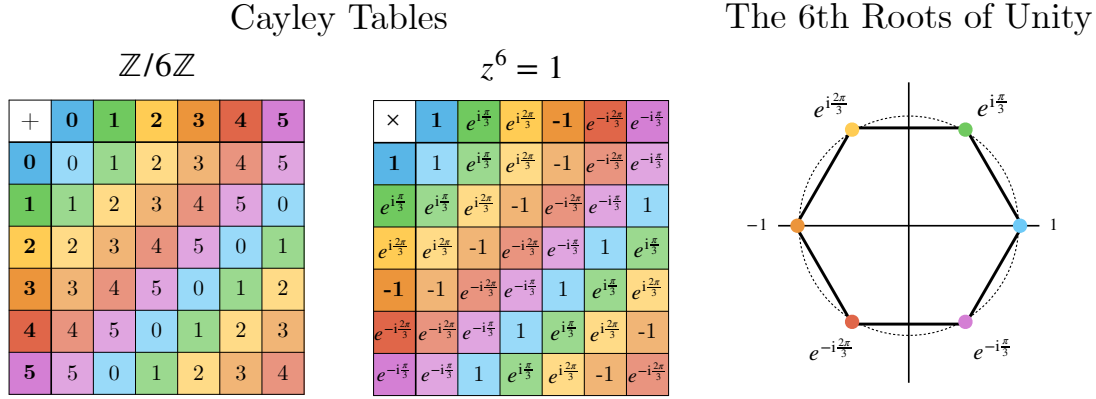


Figure 2.1: The Isomorphic Groups $(\mathbb{Z}/6\mathbb{Z}, +)$ and $(z^6 = 1, \times)$. Each cell contains the product of the pair of group elements specified by each row and column. The isomorphism between these groups is evident in the coloring of cells in the table; group elements with identical structural roles share the same color.

And lastly, every integer has an inverse (its negative, e.g. $+1$ and -1), which when summed will return the identity 0. Thus, we can conclude: $(\mathbb{Z}, +)$ is a group.

At first glance, this example may seem far removed from the concept of a *transformation*, but a closer look reveals a deep connection between familiar number systems and certain geometric transformations. Indeed, one of the insights given to us by group theory is a more general way to think about numbers as *actions*. Each integer n , for example, can be thought to specify a *translation* or *shift* on the number line to the left ($-n$) or right ($+n$). From this perspective, the group of integers under addition shares the general structural form of a familiar geometric transformation. This is more precisely formalized in the concept of an **isomorphism**.

2.2 Isomorphisms and Homomorphisms

Isomorphisms identify *structural similarities* between algebraic objects. Two groups $(G, *)$ and $(H, \odot)$ are isomorphic if there is a one-to-one, invertible correspondence between elements of the groups that respects the group operation. Concretely, a **group isomorphism** is a bijective map $f : G \rightarrow H$ such that:

$$\begin{aligned} f(u * v) &= f(u) \odot f(v), \\ f^{-1}(u \cdot v) &= f^{-1}(u) * f^{-1}(v), \end{aligned}$$

for all $u, v \in G$. The consequence is that there is exactly one element in H corresponding to each element in G . Intuitively, an isomorphism picks out pairs of corresponding elements in G and H that fulfill the same structural roles. Importantly, for f to be an *isomorphism*, it must be *bijective*; a structure-preserving map that is *not* bijective belongs to the more general class of **homomorphisms**, which may be injective or surjective. In the special case in which we define an isomorphism between a group and itself, i.e. $G = H$ and $*$ = $\odot$, this is called an **automorphism**. Some groups have interesting, non-trivial automorphism structure—additional symmetries within the group itself.

To illustrate the concept of isomorphism, we will consider two cyclic groups. The first is the *integers modulo 6* under addition, commonly denoted $(\mathbb{Z}/6\mathbb{Z}, +)$. The second is the set of complex numbers that define the *6th roots of unity*, i.e. $z \in \mathbb{C}$ such that $z^6 = 1$: $\{1, e^{i\frac{\pi}{3}}, e^{i\frac{2\pi}{3}}, -1, e^{-i\frac{2\pi}{3}}, e^{-i\frac{\pi}{3}}\}$, with $i = \sqrt{-1}$. Finite groups such as these can be easily visualized through what is called the group's **Cayley table**. A Cayley table displays all pairwise products of group elements, which are identified with rows and columns. Figure 2.1 shows the Cayley tables of $(\mathbb{Z}/6\mathbb{Z}, +)$ and $z^6 = 1$.

The isomorphism between these groups is highlighted by the coloring of cells in the table. Group elements in $(\mathbb{Z}/6\mathbb{Z}, +)$ and $(z^6 = 1, \times)$ with identical structural roles share the same color. For example, the identity element 0 in $(\mathbb{Z}/6\mathbb{Z}, +)$ maps to the identity element 1 in $(z^6 = 1, \times)$. A plot of the elements of the 6th roots of unity in the Cartesian plane (Figure 2.1, right) makes clear the geometric nature of the groups: both groups capture the action of rotation around a ring discretized into six increments. Perhaps even more explicitly geometric, both groups describe the *rotational symmetries of a hexagon*: the set of rotations of a hexagon that leave its shape invariant.

2.3 Commutativity

In these examples, the order in which the elements of the group are combined does not impact the output. However, this is not generally true of all groups, and the property defines an important distinction between classes of groups. A group is **commutative** if the order of operations does not matter, i.e. $ab = ba$. Such groups are also commonly referred to as *abelian* groups (after the mathematician Niels Abel). A group is **non-commutative** or *non-abelian* if, for at least some of the group elements, a change in the order of operations changes the result, i.e. if $ab \neq ba$ for some $a, b \in G$.

We can observe a simple example of a non-commutative group by considering the set

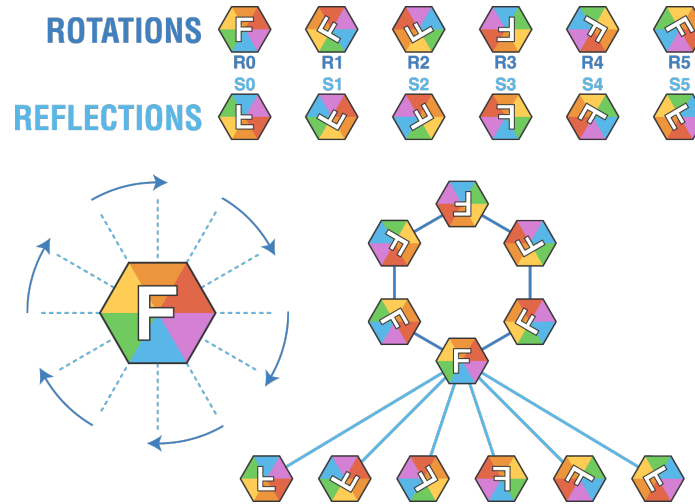


Figure 2.2: The Dihedral Group D_{12} . The twelve rotations $R_0, \dots, R_5$ and reflections $S_0, \dots, S_5$ that leave the shape of a hexagon invariant define the Dihedral Group D_{12} , the symmetry group of the hexagon. Diagram from hexnet.org

of *all* symmetries of the hexagon: all rotations *and reflections* that leave the shape invariant. This is called the Dihedral Group of Order 12, or D_{12} . Whereas the set of rotations of a hexagon define a commutative group, rotations and reflections interact in interesting ways that depend on the order in which they are applied.

Figure 2.2 depicts the 12 elements of this group: the six rotations labeled $R_0, \dots, R_5$, and six reflections labeled $S_0, \dots, S_5$. Figure 2.3 expands all pairs of these group elements into a visual Cayley table, where each [row a , column b] pair denotes the ordered product ab . The importance of the ordering of rotations and reflections can be observed in the asymmetry of the table. For example, the element corresponding to R_2S_0 is not equal to the element corresponding to S_0R_2 .

2.4 Lie Groups

Thus far, the examples we have considered have all involve finite sets of discrete transformations. Many of the geometric transformations relevant for perception are more naturally described in terms of *continuous* transformations. *Lie theory* was developed by the 19th century mathematician Sophus Lie to extend the theory of finite groups to the kinds of continuous symmetries arising from problems in geometry and dynamics.

A **Lie group** is a group that is also a differentiable manifold. Much more could

	R0	R1	R2	R3	R4	R5	S0	S1	S2	S3	S4	S5
R0												
R1												
R2												
R3												
R4												
R5												
S0												
S1												
S2												
S3												
S4												
S5												

Figure 2.3: A Visual Cayley Table of the Dihedral Group D12. The six rotations of the hexagon are labeled $R0, \dots, R5$, and the six reflections are labels $S0, \dots, S5$. Diagram from hexnet.org

be unpacked in this definition, but we will simply note here that a **manifold** is a topological space that is locally Euclidean, such as the surface of a sphere. A simple example of a Lie group is the *circle group* $S1$ of complex numbers with unit norm under multiplication, i.e. $z \in \mathbb{C}$ such that $|z| = 1$. Note that the finite cyclic groups $(\mathbb{Z}/6\mathbb{Z}, +)$ and $(z^6 = 1, \times)$ can be considered discrete approximations of $S1$.

2.5 Matrix Groups

Many of the interesting Lie groups have *matrices* as their elements, rather scalars. A **matrix group** is a group of matrices specified over a field K , with the operation of matrix multiplication. As an example, consider the set of real orthogonal matrices of dimension two under the operation of matrix multiplication, i.e. $(\mathbb{R}^{2 \times 2}, \cdot)$. This continuous Lie group is called the Special Orthogonal Group of dimension two, or $SO(2)$, and it defines the set of continuous 2-dimensional rotations.

One might wonder whether there is an isomorphism between $SO(2)$ and $\mathbb{S}^1$, as both describe the action of 2-dimensional rotation in the plane. Indeed there is; one can define the bijective map $f : SO(2) \rightarrow \mathbb{S}^1$:

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \mapsto e^{i\theta},$$

which defines an isomorphism between the two groups, using the fundamental connection between trigonometric functions and the complex exponential that Euler made explicit in his famous formula.

2.6 Group Actions

In general, we are interested in how these abstract groups *act* on the spaces over which a sensory signal is defined. Suppose we have a *sensory image*¹ that we want to transform according to some group. Mathematically, we define a sensory image as a function that maps 2-dimensional coordinates to scalar real values. That is, it is the map $I : \mathbb{R}^2 \rightarrow \mathbb{R}$. A group, such as $SO(2)$, acts on the space $\mathbb{R}^2$ to rotate the coordinates over which the signal is defined. This results in rotated signal when viewed in the original coordinates.

The technical term for this concept is the **group action**. The intuitive idea is that a group action specifies how a group element g maps elements of a space M to other elements in M . For instance, a particular 2×2 rotation matrix in $SO(2)$ maps a vector in $\mathbb{R}^2$ to another vector in $\mathbb{R}^2$, rotated with respect to the original point. Concretely, a group action is a map $\alpha : G \times M \rightarrow M$ that maps (g, m) pairs to elements of M . We say a group G *acts* on a space M if the following properties of the action α hold:

1. The identity element e_G of G maps elements of $m \in M$ to themselves, i.e. $\alpha(e_G, m) = m$
2. Two group elements g and $h \in G$ can be combined before or after the map to yield the same result, i.e. $\alpha(gh, m) = \alpha(g, \alpha(h, m))$

For simplicity, we will use the shortened notation $\alpha_g m$ to denote $\alpha(g, m)$.

Given a point $m \in M$, it can be useful to specify the set of “everything that can be reached from m by an action of something in G .” This is called the **orbit** of m , and is defined formally as the set $\{y = \alpha_g m \mid g \in G\}$. We can use this to formalize the

¹Using the word *image* in the sense of signal arising from a camera, *not* its mathematical sense.

notion of “all transformed versions of a signal.” For example, the orbit of a sensory image under the group $SO(2)$ is the set of all rotated versions of that image.

2.7 Homogeneous Spaces

The concept of a group action allows us to define another important concept: that of a **homogeneous space**. We say that M is a homogeneous space for a group G if G acts *transitively* on M —that is, if for every pair $m_1, m_2 \in M$ there exists an element of $g \in G$ such that $\alpha_g m_1 = m_2$, with α the group action on M . A homogeneous space M equipped with the action of G is called a G -space.

The concept can be clearly illustrated by considering the surface of a sphere, the space $\mathbb{S}^2$. The group $SO(3)$ of special orthogonal 3×3 matrices defines the group of 3-dimensional rotations. The sphere $\mathbb{S}^2$ is a homogeneous space for $SO(3)$. For every pair of points on the sphere, one can define a 3D rotation matrix that takes one to the other.

2.8 Invariance and Equivariance

We are now prepared to precisely define two concepts that are essential to a mathematical description of sensory perception: **invariance** and **equivariance**. Given a function Φ over a linear G -space for some group of transformations G :

Φ is *G -invariant* if $\Phi(x) = \Phi(\alpha_g x)$, for all $g \in G$,

Φ is *G -equivariant* if $\Phi(\alpha_g x) = \gamma_g \Phi(x)$, for all $g \in G$,

where α and γ are group actions on the domain and image of Φ , respectively. Intuitively, a group-invariant function returns a constant value when the space is acted on by the group—that is, it is constant on orbits. A group-equivariant function commutes with the group action on the input space. That is, applying a transformation to the input space and then computing the function produces the same result as computing the function and then applying the transformation. We will return to these definitions in Section 2.2, when we consider desiderata for sensory representations.

2.9 Representations

Groups are inherently abstract objects; a group can be defined over any arbitrary set of symbols. For sensory processing, we want to deal with objects that are natural

to compute with, such as vectors and matrices. Thankfully, the field of *representation theory* provides methods for making groups more concrete, by mapping abstract groups into the objects of linear algebra.

A **matrix representation** of a group G is map $\rho : G \rightarrow GL(n, \mathbb{C})$ that assigns elements of G to elements of the group of invertible matrices $\mathbb{C}^{n \times n}$ such that $\rho(g_1 g_2) = \rho(g_1) \rho(g_2)$. That is, ρ is a *group homomorphism*.

A group may have many representations, and many of those representations may map to objects that are equivalent under a change of basis. We say that a representation is **reducible** if there exists a change of basis that decomposes the matrix into a direct sum of other representations. An **irreducible representation** is one that cannot be further reduced in this manner. All group representations are either reducible or can be decomposed into a direct sum of irreducibles, by definition.

As an example, we will define the representations of the now-familiar group $(\mathbb{Z}/n\mathbb{Z}, +)$, which describes discrete rotations on a ring, or cyclic translations. Each integer k can be mapped to the scalar (1×1 matrix) e^{ik} such that $\rho(k_1) \rho(k_2) = e^{ik_1} e^{ik_2} = e^{i(k_1+k_2)}$. This holds for any pair of integers in the group, so ρ is a representation. As a scalar-valued representation, it cannot be reduced, so it is also an irreducible representation.

The irreducible representations of a group can be thought of as the atomic elements of the group, and they can be used to investigate properties of the group itself. The **characters** of the irreducible representations are particularly useful here. The character of an irreducible representation is a function that maps group elements to the trace of its corresponding matrix in the representation space. Characters encode many important properties of a group, and thus can be used to study its structure. Notably, the irreducible representations of all commutative groups are scalar-valued—i.e. 1×1 matrices—and thus they are equal to their characters. For non-commutative groups, representations map to matrices more generally; thus, characters offer more compact descriptions.

An important notational comment: In the fields of machine learning and neuroscience, the word *representation*—with an entirely different definition—is unfortunately both very common and very useful. At many points throughout this thesis, I will need to refer to patterns of activations in a network of biological or artificial neurons as a *representation*. From this point forward, I will use the term *neural representation* to denote the concept, and will leave *representation* for its mathematical sense.

A Representation Theoretic View of the Discrete Fourier Transform Matrix

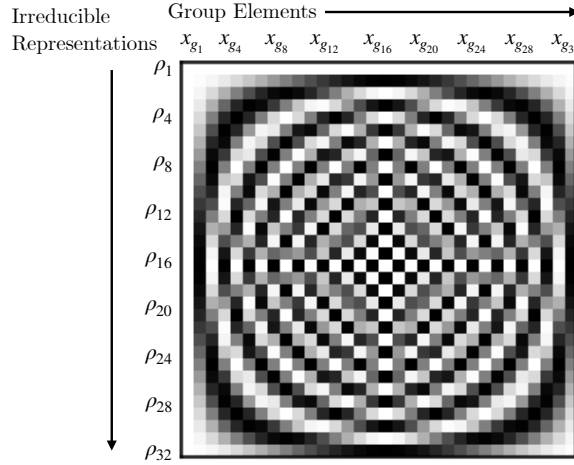


Figure 2.4: A Representation Theoretic View of the Discrete Fourier Transform Matrix. The matrix of basis vectors used to compute a discrete 1D Fourier transform defines the set of irreducible representations of the group $(\mathbb{Z}/n\mathbb{Z}, +)$. Each row of the matrix indexes a different representation $\rho_1, \dots, \rho_n$. The values of each representation $\rho(x)$ at $x \in X$, with X a G -space for $(\mathbb{Z}/n\mathbb{Z}, +)$, are computed for each column of the matrix.

2.10 Harmonic Analysis

Someone with a background in signal processing might at this point notice an important connection between the irreducible representations of the group $(\mathbb{Z}/n\mathbb{Z}, +)$ and the **Fourier transform**. The discrete Fourier transform of a signal f on a space X is defined:

$$\hat{f}_k = \sum_{x=0}^{n-1} e^{-i2\pi kx/n} f(x). \quad (2.1)$$

From a group theoretic perspective, a Fourier transform is a projection of a signal onto the irreducible representations the group $(\mathbb{Z}/n\mathbb{Z}, +)$ of cyclic translations, indexed by integer frequencies k (Figure 2.4). This gives it the property of *equivariance* to translation in the input, commonly known as the time-shift and frequency-shift properties. Intuitively, this means that translating the input and then taking the Fourier transform is equivalent to taking the Fourier transform and then phase shifting the output.

We expand the shift property of the Fourier transform here for illustration. Define the translate of f by z as $f^z(x) = f(x - z)$. Let $h = x - z$. Then, The Fourier transform of $f^z(x)$ shifts as follows:

$$\begin{aligned}
 \hat{f}_k^z &= \sum_{x=0}^{n-1} e^{-i2\pi kx/n} f(x - z) \\
 \hat{f}_k^z &= \sum_{h=0}^{n-1} e^{-i2\pi k(z+h)/n} f(h) \\
 \hat{f}_k^z &= \sum_{h=0}^{n-1} e^{-i2\pi kz/n} e^{-i2\pi kh/n} f(h) \\
 \hat{f}_k^z &= e^{-i2\pi kz/n} \sum_{h=0}^{n-1} e^{-i2\pi kh/n} f(h) \\
 &= e^{-i2\pi kz/n} \hat{f}_k.
 \end{aligned}$$

That is, the shift is proportional to the translation z , by a factor of $e^{\frac{-i2\pi}{n}zk}$. Another way of thinking of the Fourier transform is as a homomorphism that maps group actions in the space domain (cyclic translation) to group actions in the frequency domain (complex multiplication). This property of equivariance makes it useful for defining *invariant* signal transforms as well, which we return to in Sections 2.2 and 2.3.

Fourier analysis can be generalized to other compact groups, such as 2D and 3D rotation, by preserving this defining property of equivariance to the group action [111]. The (continuous) **generalized Fourier transform** is defined:

$$\hat{f}_k = \int_G \rho_k(g) f(g) dg, \quad (2.2)$$

with the set $\{\rho_k\}_{k \in G}$ defining a set of irreducible representations, G denoting a homogeneous G -space for that group, and the integral using the Haar measure dg on G . An interesting example is the spherical harmonic transform, which is a generalization of the Fourier transform to the non-commutative group of spherical rotations $SO(3)$ acting on the 2-sphere. The spherical harmonic transform projects a signal defined on a sphere onto the orthogonal basis of spherical harmonics, pictured in Figure 2.5 and defined:

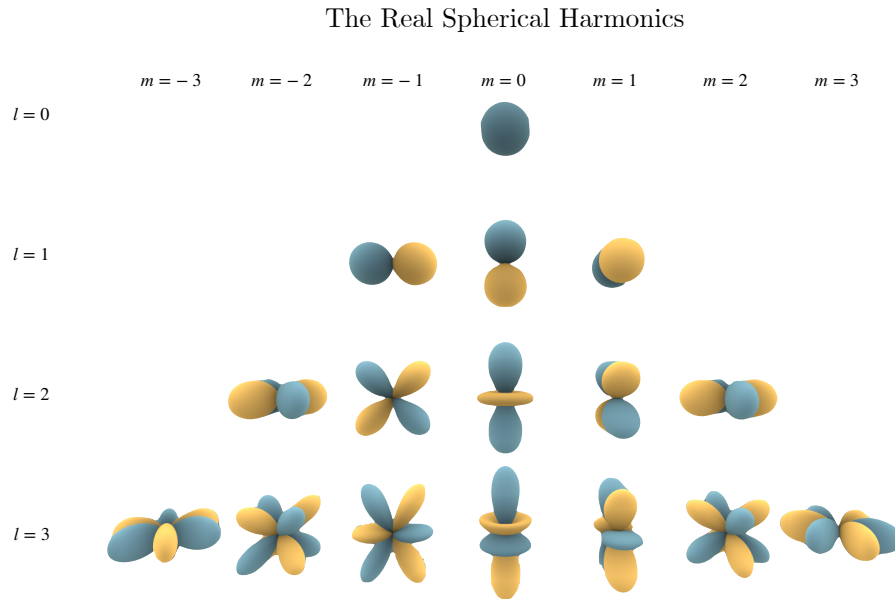


Figure 2.5: The Real Spherical Harmonics. The spherical harmonics are a generalization of sinewaves defined on the surface of the 2-sphere S^2 and describe its vibrational modes. Just like the 1D Fourier basis, they define an orthogonal basis for any function defined in spherical coordinates and can be used to decompose a spherical function into a set of coefficients over the “fundamental frequencies” of the sphere. Spherical harmonics Y

$$Y_\ell^m(\theta, \phi) = \sqrt{\frac{2\ell + 1}{4\pi} \frac{(\ell - m)!}{(\ell + m)!}} P_\ell^m(\cos \theta) e^{im\phi}, \quad (2.3)$$

with (θ, ϕ) specifying spherical coordinates and ℓ and $m \in [-\ell, \ell]$ parameterizing the modes of the harmonics.

Generalized Fourier analysis provides a natural bridge between abstract group theory and signal processing. Indeed, much of what is known of signal processing in early visual cortex indicates that the brain itself makes use of this mathematics—a point that we will expand on in detail shortly. We now return to the problem of perceptual invariance, equipped with the language and tools of group theory.

Chapter 3

A Group Theoretic Framework for Neural Computation

Possibly we can aspire to a science of mind that, by virtue of the evolutionary internalization of universal regularities in the world, partakes of some of the mathematical elegance and generality of theories of that world. The principles that have been most deeply internalized may reflect quite abstract features of the world, based as much (or possibly more) in geometry, probability, and group theory, as in specific, physical facts about concrete, material objects.

Roger Shepard

Perceptual-Cognitive Universals as Reflections of the World, 2001

The premise of this thesis is that the physical laws of the natural world—the geometry and symmetries that gives rise to our sense impressions—are embodied in the computations of individual neurons in the primate brain. Versions of this idea have been put forth by prior theorists [49, 50, 117]. Here, I aim to further the development of this perspective by offering a framework of mechanistic hypotheses for how individual neurons may implement the mathematics of group theory to construct neural representations that reflect the transformation structure of the natural world.

The idea that the brain encodes regularities in the environment is, of course, an old one, and is highly related to the efficient coding hypothesis—the idea that neurons evolved to efficiently represent the statistical structure of the natural world [5]. However, while the efficient coding approach has largely focused on faithfully reproducing signal statistics [4], the emphasis here is on the algebraic and geometric properties of

the codes and their consequent utility for behavior.

In this chapter, I introduce this perspective and precisely redefine the perceptual invariance problem through a group theoretic lens (Section 3.1). I then introduce an analytical, universal solution to the problem of selective invariance—the *bispectrum*, a Fourier-based invariant that is complete (Section 3.2), and propose two hypotheses: (1) that the receptive fields of neurons in visual cortex compute the representations of groups relevant for perception to yield *group-equivariant* neural representations and (2) that there exists a class of higher-order neurons in the visual cortex that compute bispectral coefficients to achieve robust, selective *group-invariant* neural representations (Section 3.3). In Section 3.4, I review empirical evidence from the existing literature that supports these claims, and discuss predictions for future experiments.

The proposal outlined here is nascent, and the empirical data supporting the story are far from water-tight. I present them here as suggestive support to underscore the promise of this approach for guiding future neurophysiological research.

3.1 The Problem: Invariance and Completeness

We have now established the technical definitions required to precisely the problem of *perceptual invariance*. We are interested, in particular, in defining the concept of *selective invariance*—simultaneous selectivity for pattern structure and invariance to transformation structure.

We define a sensory signal $f : M \rightarrow \mathbb{R}$ as a function on a G -space M that maps elements of M to real scalar values, where G is a group that acts on M . We abstractly model sensory processing as a map $\Phi : f \rightarrow \mathbb{C}^n$ from the signal space to an n -dimensional neural representation space defined over the complex field.

If the sensory processing map Φ preserves *all* and *only all* the signal structure, independent of the actions of a group on that signal’s domain, we say it is a **complete invariant**. Concretely:

Φ is **G -invariant** if $\Phi(f(m)) = \Phi(f(\alpha_g m))$, for all $g \in G$,

Φ is **complete** if $\Phi(f(\alpha_g m_1)) \neq \Phi(f(\alpha_g m_2))$ if $m_1 \notin \{y = \alpha_g m_2 | g \in G\}$,
for all $g \in G$, for all $m_1, m_2 \in M$.

That is, a *complete* invariant maps only objects that are in the same orbit under the

group action to the same point.

One natural way to achieve invariance is to convert the group action in the signal domain to a group action in the neural representation space that can be easily eliminated. This is what the power spectrum does. Given the Fourier transform $\hat{f}$ of a signal f , the **power spectrum** is defined:

$$q_k = \hat{f}_k \hat{f}_k^\dagger, \quad (3.1)$$

where † denotes the complex conjugate. The Fourier transform converts translations in the signal domain into phase shifts in the frequency space. The power spectrum achieves invariance by eliminating the Fourier phase information entirely. By multiplying $\hat{f}$ with its conjugate, the translate factors in $\hat{f}$ and $\hat{f}^\dagger$ cancel out:

$$\begin{aligned} q_k^z &= \left(e^{-i2\pi kz/n} \hat{f}_k \right) \left(e^{-i2\pi kz/n} \hat{f}_k \right)^\dagger \\ &= e^{-i2\pi kz/n} e^{i2\pi kz/n} \hat{f}_k \hat{f}_k^\dagger \\ &= \hat{f}_k \hat{f}_k^\dagger \\ &= q_k. \end{aligned}$$

This is, unfortunately, a heavy-handed approach that eliminates *both* transformation and signal structure, thus making the power spectrum *incomplete* (See Figure 1.3 in Chapter 1). However, there *is* a Fourier-based invariant that possesses the rare property of completeness. This invariant is a third-order generalization of the power spectrum, called the **bispectrum**, and it offers a universal solution to the problem of selective invariance.

3.2 A Universal Solution: The Bispectrum

The bispectrum is a lesser known Fourier invariant that is *complete*, which means that it collapses *all and only all* orbits of a signal under a group action to the same point [63]. It achieves this by retaining relative phase structure while eliminating absolute phase. The bispectrum is parameterized by pairs of frequencies in the Fourier basis. Intuitively, bispectral coefficients indicate the strength of couplings between different phase components in the data. This preserves the relative phase structure

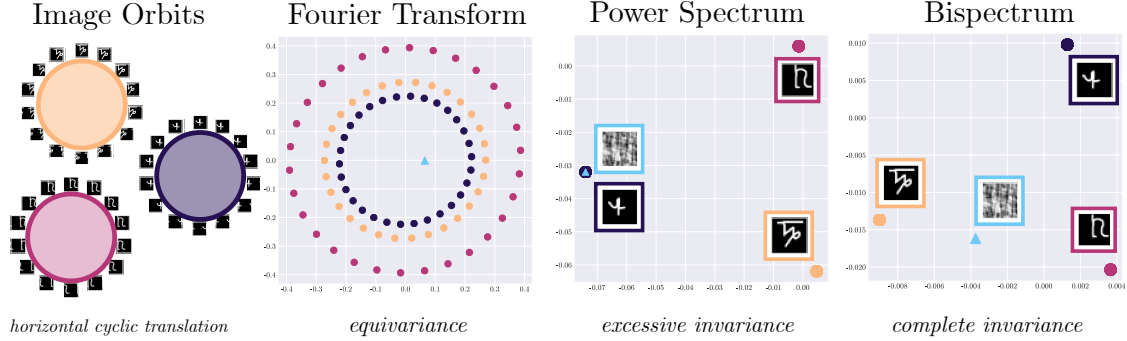


Figure 3.1: The Bispectrum is a Complete Invariant. (Left) Three hand-written characters and their corresponding orbits under one dimension of cyclic translation. (Middle Left) All samples mapped through the Fourier transform, then plotted in the plane formed by the first two principle components. (Middle Right) The power spectrum collapses the orbits of each character, successfully achieving invariance to translation. However, it is *excessively* invariant. A phase-scrambled image (light blue) maps to the same point as the purple digit class because it has the same power spectrum. (Right) The bispectrum is a complete invariant, mapping *all and only all* points that are identical up to the group action to the same point.

that defines structure in signals—the structure that the power spectrum loses. For a pair of frequencies k_1 and k_2 , the bispectrum β is defined:

$$\beta_{k_1, k_2} = \hat{f}_{k_1} \hat{f}_{k_2} \hat{f}_{k_1+k_2}^\dagger, \quad (3.2)$$

with the summation $k_1 + k_2$ taken modulo the dimension of the signal. Due to the conjugated third term, any absolute phase shifts resulting from translations in the input cancel out, leaving the bispectrum invariant. Under translation, β becomes:

$$\begin{aligned} \beta_{k_1, k_2}^z &= e^{-i2\pi k_1 z/n} \hat{f}_{k_1} e^{-i2\pi k_2 z/n} \hat{f}_{k_2} e^{i2\pi(k_1+k_2)z/n} \hat{f}_{k_1+k_2}^\dagger \\ &= e^{-i2\pi k_1 z/n} e^{-i2\pi k_2 z/n} e^{i2\pi k_1 z/n} e^{i2\pi k_2 z/n} \hat{f}_{k_1} \hat{f}_{k_2} \hat{f}_{k_1+k_2}^\dagger \\ &= \hat{f}_{k_1} \hat{f}_{k_2} \hat{f}_{k_1+k_2}^\dagger \\ &= \beta_{k_1, k_2}. \end{aligned}$$

The third term in the bispectrum can be thought of as a kind of normalization factor that divides out the phase changes in frequency space due to a translation in the input, thus putting it into a “canonical” reference frame. Unlike the power spectrum, the bispectrum can be used to reconstruct the original signal up to translation, as it

Symmetries of the Triple Correlation and Bispectrum

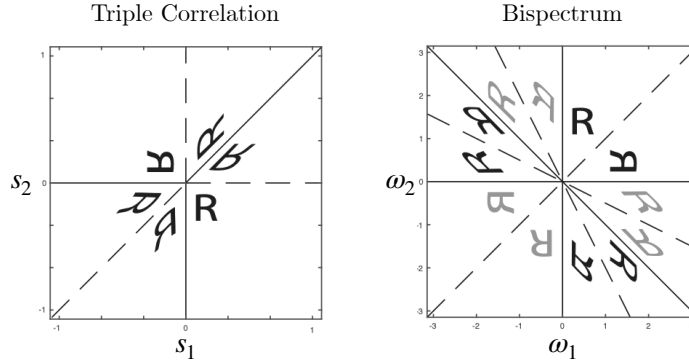


Figure 3.2: Symmetries of the Triple Correlation and Bispectrum. For a real-valued signal, the six-fold symmetry of the triple product becomes a twelve-fold symmetry in which six pairs of regions are conjugate symmetric (with the complex conjugate indicated in grey). Adapted from [91].

encodes the patterns of relative phase relationships, while removing absolute phase (Figure 3.1). These relative phase relationships define much of the structure in a signal and are particularly important in defining edges and contours (Figure 1.3). Being complete, the bispectrum also preserves the magnitude spectrum of the signal, which comprises the subset of the bispectral coefficients for which $k_2 = 0$.

The bispectrum has n^2 terms (with n = number of frequencies). However, the object is highly redundant and is reducible to $n + 1$ terms without loss of information [68]. This is due to the following symmetries, depicted in Figure 3.2. For $\omega_1 = 2\pi k_1$:

$$\begin{aligned}
 \beta_{\omega_1, \omega_2} &= \beta_{\omega_2, \omega_1} \\
 &= \beta_{-\omega_2, -\omega_1} = \beta_{-\omega_1 - \omega_2, \omega_2} \\
 &= \beta_{\omega_1, -\omega_1 - \omega_2} = \beta_{-\omega_1 - \omega_2, \omega_1} \\
 &= \beta_{\omega_2, -\omega_1 - \omega_2}.
 \end{aligned}$$

Thus, the subset of bispectral coefficients for which $\omega_2 \geq 0$, $\omega_1 \geq \omega_2$, $\omega_1 + \omega_2 \leq \pi$ is sufficient for a complete description of the bispectrum.

Historically, the bispectrum emerged from the study of the higher-order statistics of non-Gaussian random processes [132, 9, 91]. Both the power spectrum and bispectrum are intimately related to the second- and third-order statistics of a signal, as the

Fourier transforms of the two- and three-point autocorrelation functions, respectively. The two-point **autocorrelation** A_2 of a complex-valued function f on the real line is the integral of that function multiplied by a shifted copy of it:

$$A_{2,f}(s) = \int_{-\infty}^{\infty} f^\dagger(x) f(x+s) dx, \quad (3.3)$$

where s denotes the increment of the shift. Taking its Fourier transform, it becomes:

$$\hat{A}_{2,f}(s) = \hat{f}_k \hat{f}_k^\dagger, \quad (3.4)$$

which is the power spectrum. The three-point autocorrelation, also called the **triple correlation**, is the integral of a function multiplied by two independently shifted copies of it:

$$A_{3,f}(s_1, s_2) = \int_{-\infty}^{\infty} f^\dagger(x) f(x+s_1) f(x+s_2) dx. \quad (3.5)$$

Taking its Fourier transform it becomes:

$$\hat{A}_{3,f}(s_1, s_2) = \hat{f}_{k_1} \hat{f}_{k_2} \hat{f}_{k_1+k_2}^\dagger, \quad (3.6)$$

which is the bispectrum. The triple correlation has zero expected value for a Gaussian process. Consequently, the bispectrum has been used as a tool for measuring the non-Gaussianity of a signal. The later work of Yellott and Iverson [137] established that every integrable function with compact support is completely identified—up to translation—by its bispectrum, thus establishing the bispectrum as a unique and complete signature of signal structure.

Ramakrishna Kakarala is to credit for defining the more general, group-theoretic form of the bispectrum, making use of the generalized Fourier transform to define bispectra that are invariant to arbitrary compact groups [65, 62, 64]. The **generalized bispectrum** for commutative groups is defined over a pair of irreducible representations ρ_1, ρ_2 :

$$\beta_{\rho_1, \rho_2} = \hat{f}_{\rho_1} \hat{f}_{\rho_2} \hat{f}_{\rho_1 \rho_2}^\dagger. \quad (3.7)$$

The irreducible representations (irreps) of commutative groups map to scalar values. However, for *non-commutative* groups, irreps map to matrices, more generally. Any pair of irreps can be combined through the tensor product to yield another representation. This representation can be decomposed into a direct sum of irreps through the Clebsch-Gordan decomposition:

$$\rho_1(g) \otimes \rho_2(g) = C_{\rho_1, \rho_2} \left[\bigoplus_{\rho \in \rho_1 \otimes \rho_2} \rho(g) \right] C_{\rho_1, \rho_2}^\dagger, \quad (3.8)$$

where C is a Clebsch-Gordan matrix, $\otimes$ is the tensor product, and $\oplus$ is a direct sum over irreps. The unitary Clebsch-Gordan matrix C is analytically defined for each pair of representations. The Clebsch-Gordan decomposition is well-known in quantum mechanics, as it is central to the algebra of angular momentum. With this in mind, the generalized form for non-commutative groups can be written:

$$\beta_{\rho_1, \rho_2} = [\hat{f}\rho_1 \otimes \hat{f}\rho_2] C_{\rho_1, \rho_2} \left[\bigoplus_{\rho \in \rho_1 \otimes \rho_2} \hat{f}_\rho^\dagger \right] C_{\rho_1, \rho_2}^\dagger. \quad (3.9)$$

Kakarala went on to prove that the generalized bispectrum in both its commutative and non-commutative form is *complete* [63]. This makes it a universal solution to the selective invariance problem, for *all compact groups*.

The generic translation-invariant bispectrum has been employed in the context of image statistics by Zetsche and collaborators [71]. It has appeared in its generalized, group theoretic form in the works of Kakarala and Kondor, which have used the $SO(3)$ -invariant bispectrum for invariant signal processing of functions defined on the surface of a sphere [69, 64]. In prior work, I and collaborators used the S_n permutation-invariant bispectrum to construct invariant representations of structured graphs such as molecules and proteins [115]. These results demonstrate the bispectrum as a powerful and general computational approach to invariant signal processing. I suggest that it may also shed light on how the brain achieves selective perceptual invariance.

3.3 Proposal: Group Representations as Universals

Given the unique and universal properties of the generalized bispectrum, it can be considered an optimal solution to the problem of selective invariance. Moreover, given that it is Fourier-based, it can be plausibly implemented in the visual system, for which there is considerable evidence of Fourier-based neural computations [102, 80, 25]. Here, I propose the following two hypotheses:

- **Equivariance:** The receptive fields of neurons in the visual cortex compute the localized representations of groups relevant for perception, resulting in group-equivariant neural representations, and allowing selective perceptual invariants to be computed through simple computations in the neural representation space.
- **Invariance:** Higher-order neurons in the visual cortex combine the outputs of group-equivariant neurons to compute group-invariant bispectral coefficients and achieve robust, selective invariance to the actions of groups on perceptual objects.

The classical oriented receptive fields of V1 neurons have received many functional interpretations since their discovery, including edge detection [56], spatial frequency analysis [102], and sparse coding [94]. I propose an alternative interpretation of oriented receptive fields as *localized representations of the group of 2D cyclic translation* $\mathbb{S}^1 \times \mathbb{S}^1$. This reinterpretation emphasizes an important functional role: *equivariance to translation*. Moreover, it suggests a generalization of classical Gabor receptive fields as special cases of a more general strategy for computing the transformation structure in visual information with group representations.

This group theoretic generalization provides a richer interpretation of simple and complex cell receptive fields and suggests empirical predictions for a class of more general *group theoretic receptive fields*. Extending the group theoretic framing of the Gabor model of simple and complex cell receptive field properties, I hypothesize the more general existence of local group-representation-selective cells that are *equivariant* to the actions of groups relevant for perception, such as 2- and 3-D rotation, scaling, shearing, and certain continuous deformations (Figure 3.3). In addition, I predict the existence of higher-order *bispectral neurons* in the visual cortex that compute *group-invariant* bispectral coefficients from the outputs of simpler group-equivariant neurons.

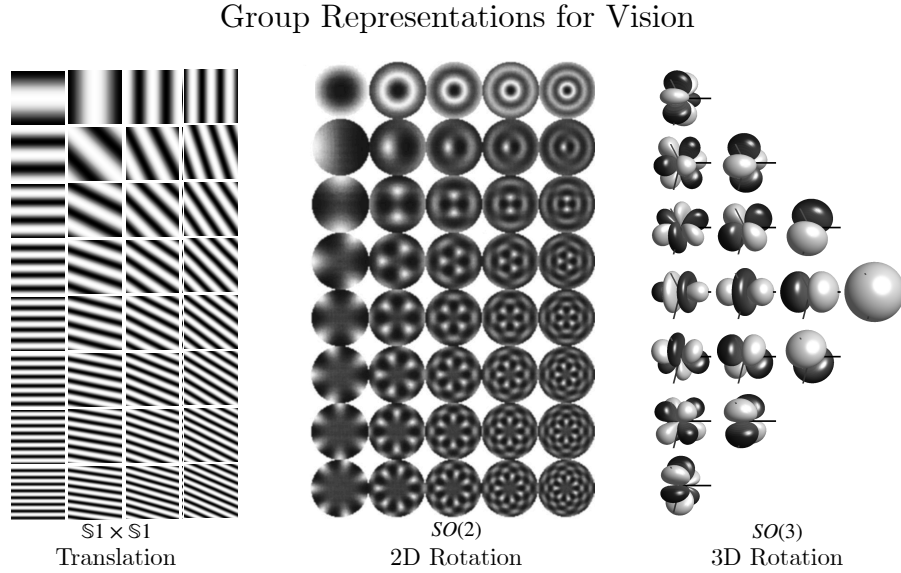


Figure 3.3: Group Representations for Vision. The representations of groups such as $\mathbb{S}1 \times \mathbb{S}1$, $SO(2)$, and $SO(3)$ may allow individual neurons in the visual system to compute group-equivariant transformations of signals.

3.4 Evidence in Biology

The above proposal is primarily intended as a set of hypotheses to guide future empirical work. However, there is some evidence in the existing literature that supports these claims, which we review now.

3.4.1 Equivariance: Group Representations in Visual Cortex

The most straightforward phenomena consistent with this hypothesis are the response properties of simple cells, whose receptive fields are naturally modeled as linear Gabor filters [80]. A Gabor filter is a 2-dimensional Fourier filter that has been localized with a Gaussian window. Just as the 1-dimensional Fourier basis defines the set of irreducible representations for the group of 1-dimensional cyclic translation $\mathbb{S}1$, the 2-dimensional Fourier basis defines the set of irreducible for 2D cyclic translation $\mathbb{S}1 \times \mathbb{S}1$. The 2D Fourier basis can also be described as the harmonics of the torus $\mathbb{T}^2$, which is a homogeneous space for this group.

The receptive fields of simple cells in V1 have received many functional interpretations through the decades. The proposal set forth here situates simple cells as part of a broader class of neurons that compute the representations of the groups relevant for perception, with receptive fields characterized by the group representations depicted

in Figure 3.3, among others. As a first step towards testing this hypothesis, we are interested in identifying neurophysiological phenomena that are consistent with the proposal—i.e. neurons in the visual cortex that are selective for or equivariant to transformations such as 2D and 3D rotation and scaling, and neurons that show receptive field structure similar to those in Figure 3.3.

With or without the concept of a group, there has long been the idea in vision science that translation, rotation, and scaling are fundamental classes of visual transformations. This perhaps prompted studies by Tanaka and Saito investigating whether there exist neurons in the middle temporal area (MT) that are selective for rotating and scaling patterns of optical flow.

The existence of neurons in MT that are selective for directed *translational* motion is well-established [133]. In a series of several studies, Tanaka and Saito established the existence of additional classes of MT neurons selective for rotational and scaling flow patterns [114, 123, 124, 122]. In an 1986 single-electrode study of 422 neurons in the MT of anaesthetized monkeys, the authors found 217/422 (51.4%) neurons selective for translational movement in a particular direction, 66/422 (15.7%) neurons selective for expanding or contracting motion, and 58/422 (13.7%) neurons selective for clockwise or counterclockwise rotation in the frontoparallel plane.

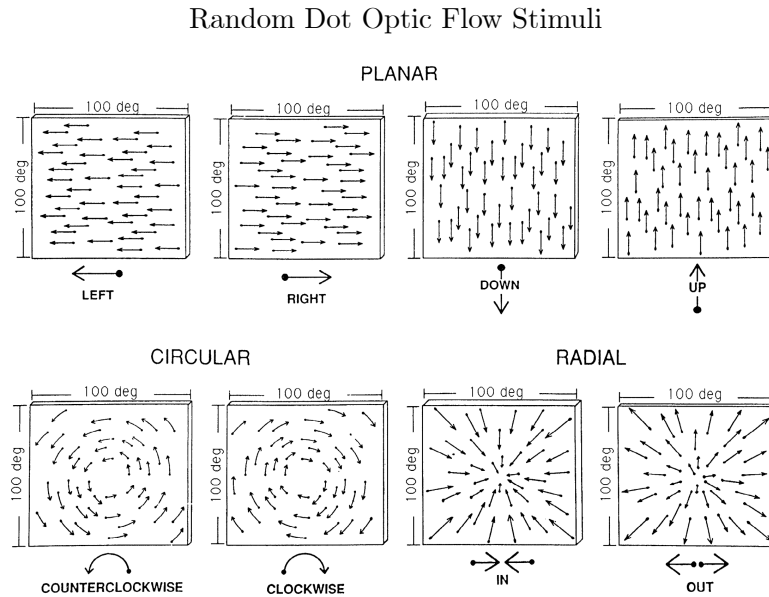


Figure 3.4: Random Dot Optic Flow Stimuli used in studies by Duffy and Wurtz (1991) [28, 29] investigating the selectivity of MT neurons for translational, rotational, and scaling motion patterns. Figure adapted from [28].

These findings were replicated in their own later studies [123, 122], as well as studies by Duffy and Wurtz (1991a, 1991b) [28, 29] and Orban, et al. (1992) [97]. Figure 3.4 shows the stimuli used in the studies by Duffy and Wurtz [28], which consist of arrays of random dots moving in planar, circular, or radial patterns.

Other studies have demonstrated the existence of neurons in V1 and V4 with receptive fields that resemble the representations of $SO(2)$ and the Lie differential operators for scaling. One study by Gallant, et al. (1993) [38] found neurons in macaque V4 selective for sinusoidal gratings in Cartesian, polar, and hyperbolic coordinates, as shown in Figure 3.5. The selectivity of neurons for the radial and concentric patterns is particularly notable, as these patterns are closely related to the Lie differential operators for rotation and scaling.

Sinusoids in Cartesian, Polar, and Hyperbolic Coordinates

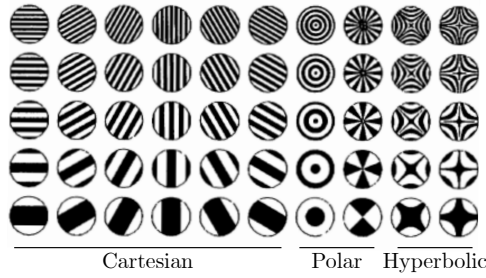


Figure 3.5: Sinusoids in Cartesian, Polar, and Hyperbolic Coordinates used as stimuli in the Gallant, et al. (1993) study of V4 receptive fields. Figure adapted from [38]

Another study by Victor et al. (2005) used a basis of two-dimensional Hermite functions in both polar and Cartesian coordinates (Figure 3.6) to probe the receptive field properties of neurons in cat and macaque V1. The study compared the responses of individual neurons to the polar and Cartesian stimuli, and found that most neurons showed significant preference for one stimulus class over the other, with many examples of selective preference for the polar-Hermite patterns. This suggests some sensitivity in V1 neurons to the polar Hermite structure, although it does not necessarily establish selectivity for these functions.

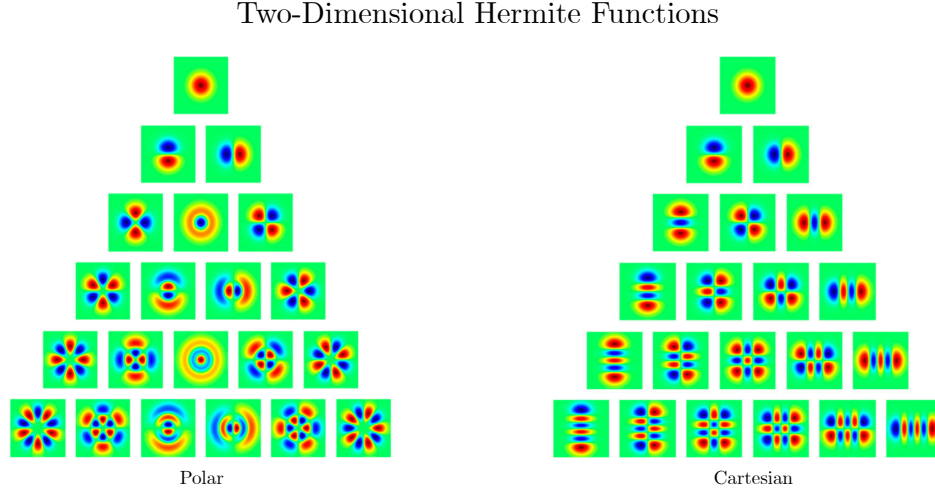


Figure 3.6: Two-Dimensional Hermite Functions in polar and Cartesian coordinates, similar to those used as stimuli in the Victor, et al. (2005) study of the receptive field properties of V1 neurons [135]. Figure adapted from [54].

A recent two-photon calcium imaging study from Tang and collaborators [125] demonstrated the existence of V1 cells in awake macaques with striking non-classical response properties: robust selectivities to geometric patterns such as targets, triangles, and whisker patterns in a dataset including more classical stimuli such as bars. In addition, many of these neurons exhibited translation and rotation invariance to the patterns they were selective for. The stimuli used in these studies (Figure 3.7, Top Right) were generated intuitively to span a range of geometric features not conventionally tested in vision science.

The revealed selectivities and invariances exhibited by these neurons suggest a much richer set of receptive field properties in V1 than classically assumed. Moreover, many of these selectivities are consistent with the group representation hypothesis. Figure 3.7 shows the responses of two example neurons from the study, in response to the complete dataset of 9,500 rotated and translated versions of the patterns depicted in the top right. The histogram shows the neurons' responses over the dataset, with the vertical red bar placed at 3 standard deviations away from the mean. All stimuli eliciting responses above this threshold are displayed below. These two neurons show remarkably consistent selectivity: Neuron 4 (top) is rotation-invariantly selective for a whisker pattern suggestive of the Lie differential operator for scaling; Neuron 1133 (bottom) is selective for a target structure suggestive of the disk harmonics, or representations of $SO(2)$.

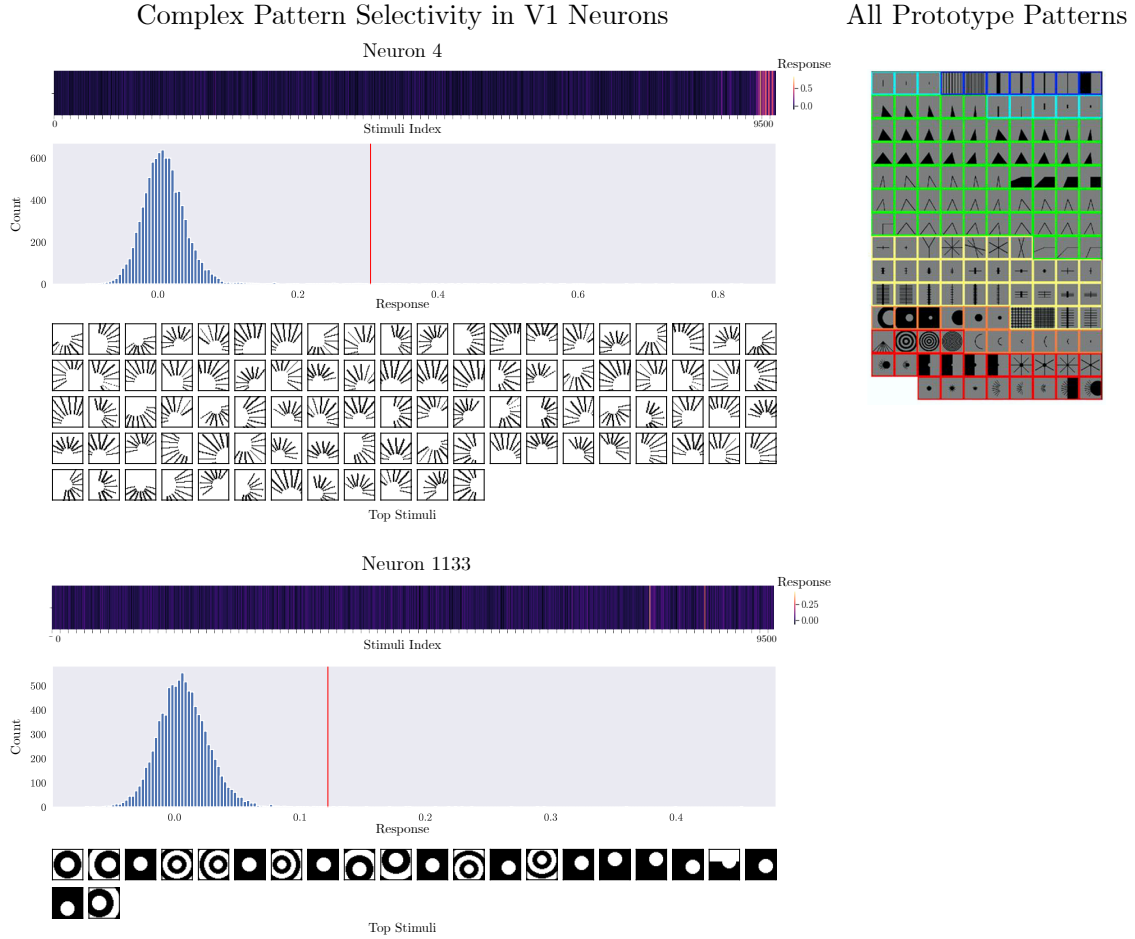


Figure 3.7: Complex Pattern Selectivity in V1 Neurons in Tang, et al. (2018) [125]. Figures on the left produced from data shared by the authors of the study, and prototype patterns reproduced from [125].

In a study of inferior temporal cortex (IT), neurons invariant and equivariant to 3D rotations have also been discovered. Logothetis, et al. (1995) [75] investigated the responses of neurons in macaque IT to images of novel, artificial 3-dimensional objects, viewed in different 3-dimensional rotations. In this study, macaques were introduced to images of novel 3D objects, such as the folded wire object depicted at the bottom right of Figure 3.8. The macaques were trained to recognize these objects independent of their pose in 3D space, through a task that required selecting one lever for the target image, and another for any distractors. Out of a population of 960 IT neurons, the authors found 61 neurons (7.66%) that selectively responded to

particular views (3D orientations) of these novel objects.

The responses of 4 such neurons are displayed in Figure 3.8 (a - d). Notably, these neurons appear to respond in an equivariant manner, with their responses continuously decreasing as objects are rotated away from the preferred orientations. Several neurons also showed simultaneous preference for two views that are 180° apart, such as neuron (c) in Figure 3.8. The tuning curves shown here are suggestive of the equivariant responses of a spherical harmonic filters. In addition, the authors found a small set of neurons that responded invariantly to all rotations of the novel objects (Figure 3.8, e).

View-Selective and Invariant Neurons in Logothetis, et al. (1995)

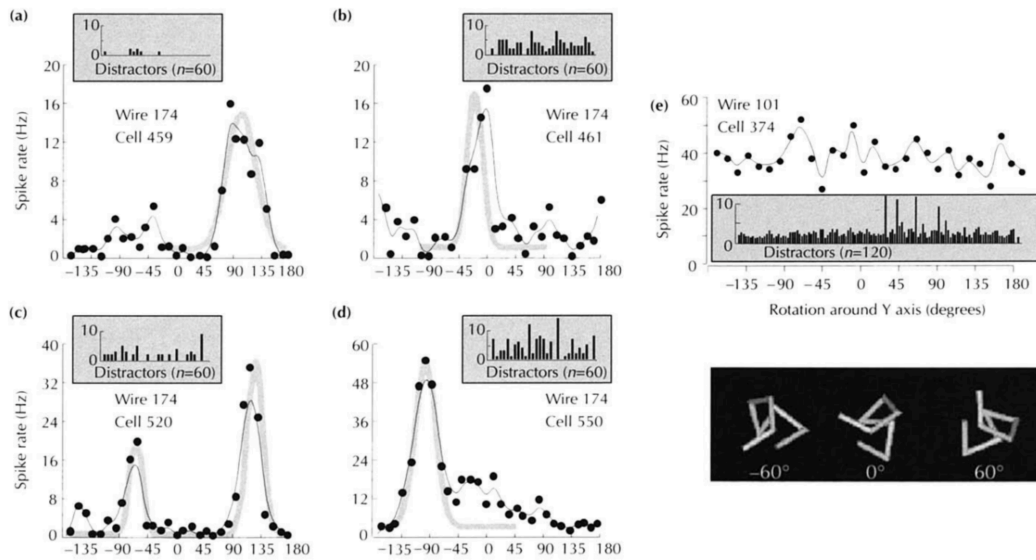


Figure 3.8: View-Selective and Invariant Neurons in Logothetis, et al. (1995). Neurons (a-d) show equivariant or view-selective responses, whereas neuron (e) fires invariantly for all views of the object. An example stimulus object under three orientations is shown in the bottom right.

The above review highlights a set of neurophysiological phenomena across visual areas in the brain—V1, V4, MT, and IT—that are consistent with the group representation hypothesis. While these data are far from conclusive evidence, they underscore the relevance and potential of this perspective for explaining extra-classical phenomena and for guiding future hypothesis-driven empirical research.

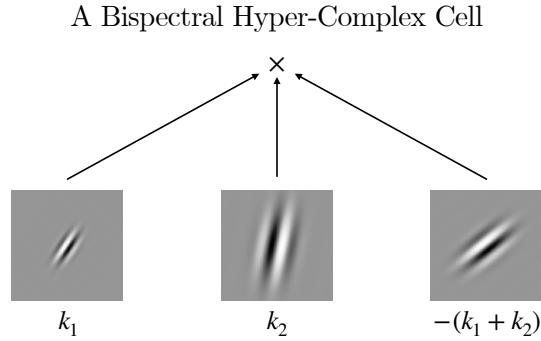


Figure 3.9: A Bispectral Hyper-Complex Cell. The outputs of three simple cells of frequencies k_1 , k_2 , and $k_3 = -(k_1 + k_2)$ are multiplied to yield a bispectral coefficient. Only the imaginary component of the receptive fields are shown here.

3.4.2 Invariance: Bispectral Hyper-Complex Cells

Classically, translation-invariant neurons in V1 (complex cells) are modeled as the amplitude of the output of a quadrature pair of translation-equivariant simple cells [1] (i.e. a localized power spectrum). While translation invariant, this operation eliminates the relative phase structure, which contains much of the important information in a signal [96]. From a theoretical perspective, it would be surprising if the robustness of primate vision were built on such a lossy operation. Empirically, there is reason to believe it is not: neurophysiological data from several studies have established that translation-invariant complex cells are indeed sensitive to relative phase structure—meaning that they preserve rather than eliminate this information [31, 46, 83, 58, 138]. These results require an explanatory model that is simultaneously invariant to absolute phase and sensitive to relative phase. The bispectrum provides exactly that explanation.

Concretely, I raise the possibility here of a class of higher-order “hyper-complex” bispectral neurons that combine the outputs of particular triplets of simple cells. To compute a bispectral coefficient, the frequencies of the neurons’ receptive fields must stand in a specific relationship to ensure absolute phase elimination: the frequency k_3 of neuron 3 must be equal to the negative of the sum of the frequencies of the other two neurons k_1 and k_2 —i.e. $k_3 = -(k_1 + k_2)$. The basic model is depicted in Figure 3.9. The output of a population of such “bispectral neurons” would simultaneously preserve relative phase structure and eliminates absolute phase, thus achieving *selective invariance*.

An interesting consequence of this model is that it replicates some of the non-linear

“extra-classical” receptive field properties observed in V1 that fall outside of the canonical simple / complex cell model. In Hubel and Wiesel’s original papers, they identified a third class of cells, which they termed *hyper-complex*, that exhibited the non-linear response property commonly referred to as *end-stopping* [55]. The response of a simple or complex cell will grow in magnitude as the length of a bar in its preferred orientation is extended. This is called length-summing, and is consistent with the Gabor simple cell and energy complex cell models. Similarly to simple and complex cells, hyper-complex cells show selectivity for the orientation and width of a bar. However, the response of a hyper-complex cell will *decline* as a bar in its preferred orientation is increased in length. This was originally interpreted by Hubel and Wiesel as a selectivity for *length*. Figure 3.10 illustrates the phenomena.

Later studies discovered additional phenomena associated with these neurons. One such example is *sign-of-contrast sensitivity*. A complex cell will give an equal response to a light bar on a dark background or a dark bar on a light background. That is, the response is invariant to sign of contrast. Hyper-complex cells behave differently. The end-stopping phenomenon in hyper-complex cells can be extinguished if the sign of the pattern (light flipped to dark or vice versa) on the ends of the bar is flipped from the center of the bar [136]. That is, the response of a hyper-complex cell will *increase* in magnitude if the preferred length bar is extended with a flipped sign intensity.

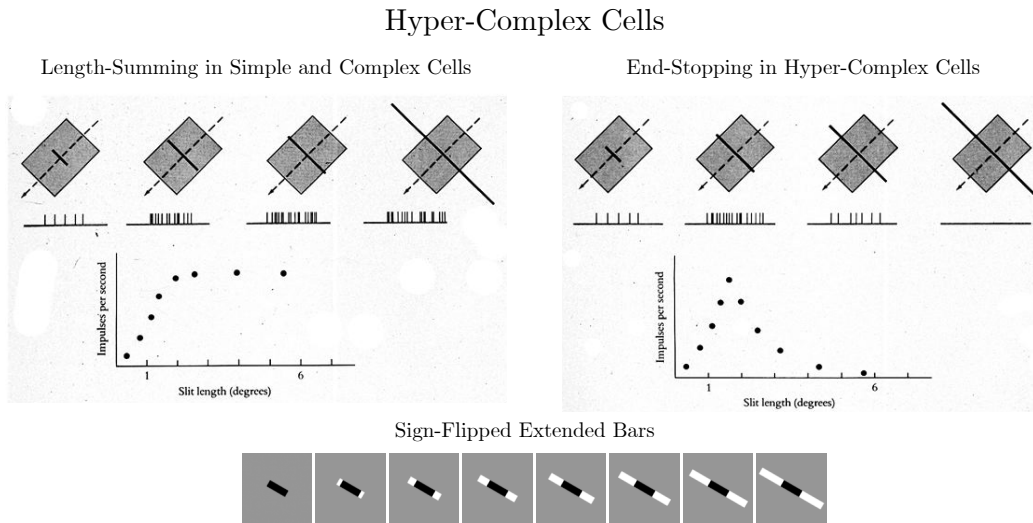


Figure 3.10: Hyper-Complex Cells show decreased responses to bars extended past a preferred length (top right) [55]. By contrast, the responses of simple and complex cells grow monotonically as a bar is increased (top left). However, the responses of hyper-complex neurons will grow in magnitude to bars increased in length by sign-flipped extensions, such as those depicted in the bottom.

The phenomenon of end-stopping has received many proposals for its underlying cause, with most models involving a recurrent inhibitory mechanism [139, 141]. Notably, all proposals require a non-linear or higher-order model. We find that the basic end-stopping phenomenon can also be explained by the bispectral neuron model. Using a bank of 440 Gabor wavelets of dimension 100×100 , with x-axis and y-axis frequencies spanning -10 to 10, we generated all non-redundant bispectral neurons, computed as the triple products of filters meeting the indexing criteria. We examined neuron response magnitudes for bars of varying lengths and orientations, testing (a) whether the neurons were orientation-selective, and (b) whether they exhibited length-summing or end-stopping behavior. Out of 54,180 bispectral neurons, we found that 50.80% of the neurons were end-stopped, 38.73% were not orientation-selective, and 10.48% were length-summing.

Figure 3.11 shows four example neurons from the model: two end-stopped neurons, one classical neuron, and one non-selective neuron. We find that bispectral neurons exhibit length-summing responses when the orientations of the receptive fields of the three simple cell inputs are aligned. When the orientations are close to orthogonal, the neuron is non-selective. When one pair of neurons is aligned and the other is separated by 45° , the neurons exhibit end-stopping.

Moreover, the model also replicates the phenomenon of sign-of-contrast sensitivity. Figure 3.12 shows how the responses of the end-stopped cells from Figure 3.11 are modulated by flipped sign bars of growing length: as the sign-flipped tips are extended the neurons' responses grow, thus extinguishing the end-stopping phenomenon. For these experiments, the orientation and length of the bar that elicited the maximum response was selected. The length of the preferred bar was then increased in size with a flipped sign intensity, and the neuron response was computed. The response to the flipped-sign stimuli is plotted in a dashed black line.

Notably, a similar result was discovered in a related, second-order model [119] that multiplicatively combines the outputs of simple cells tuned to different orientations and frequencies. The model also exhibits the end-stopping phenomenon. Indeed, we were able to replicate the result using multiplicative combinations of only two simple cells. While this second-order model is simpler in some sense, it also lacks a theoretical justification. We suggest the goal of computing selective invariants as an alternative explanation for this non-linear phenomenon.

Bispectral Neuron Response Properties

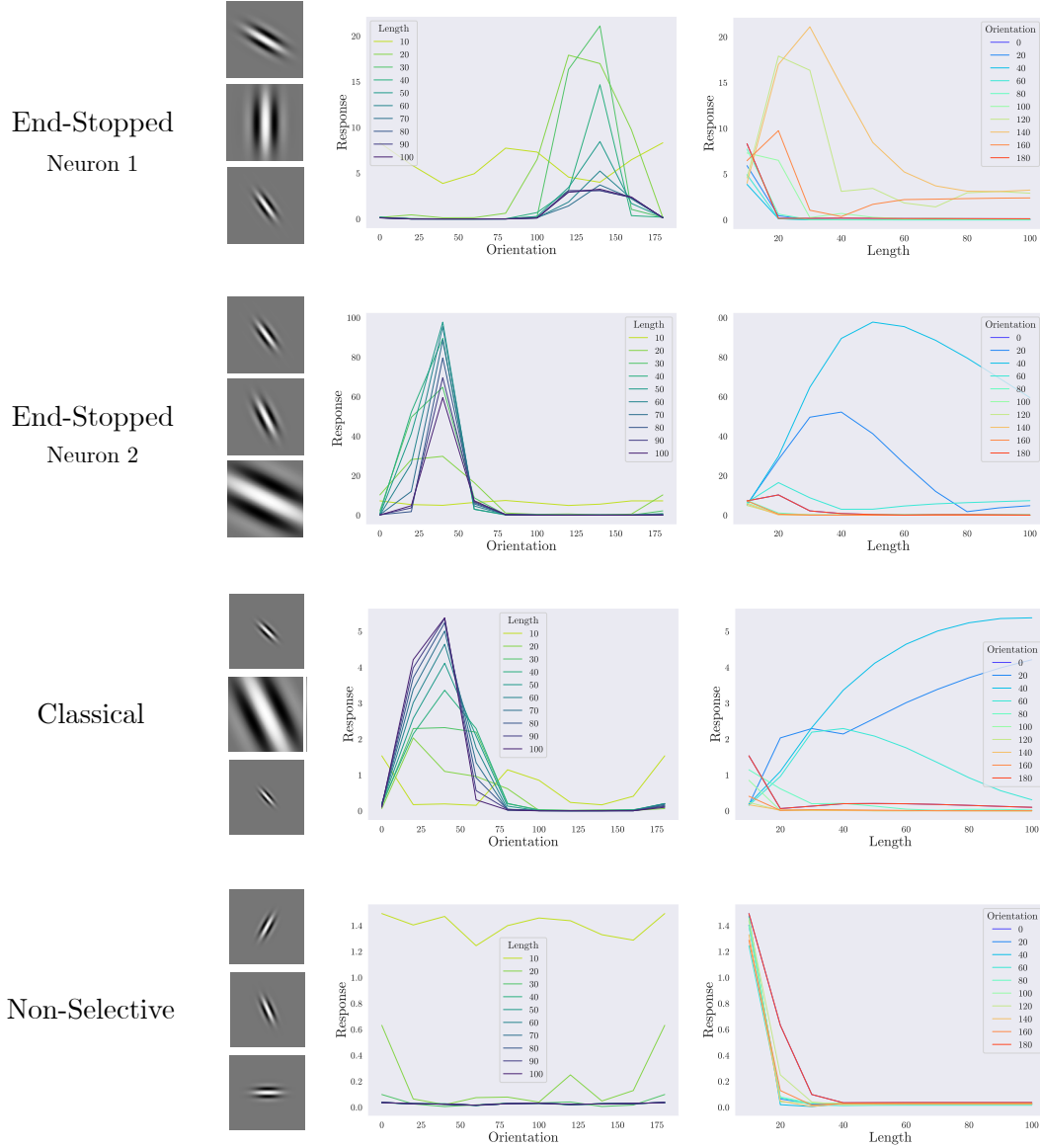


Figure 3.11: Bispectral Neuron Response Properties. Four example bispectral neurons exhibiting end-stopping (top two), classical (third), and non-selective (bottom) responses to oriented bars. The orientation of the bar is varied on the x-axis in the left plots, while the length of the bar is varied on the x-axis in the right plots.

Sign-of-Contrast Sensitivity in Bispectral Neurons

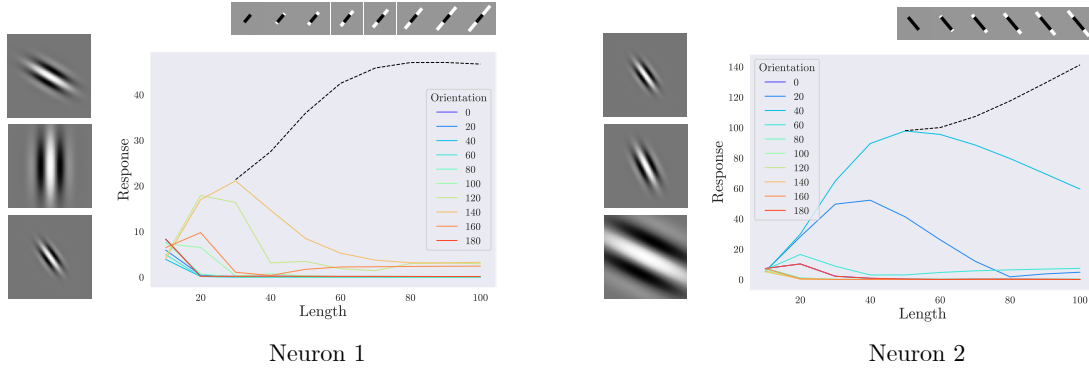


Figure 3.12: Sign-of-Contrast Sensitivity in Bispectral Neurons. Additional data are added to the length-varying plots from Figure 3.9. The black dotted line shows the neurons’ responses to optimal stimuli that have been extended with opposite sign flanks (displayed on top). These stimuli eliminate the end-stopping phenomenon, and the neurons’ responses grow as the flanks are extended.

An important challenge with the model as proposed here, is that by localizing the Fourier basis filters—essentially computing a wavelet bispectrum—we have removed the guarantees of the analytical object with respect to invariance and selectivity. Further developments are needed to establish the mathematical theory of the localized bispectrum as it may be implemented in neural machinery. We leave this for future work.

3.5 Discussion

The results reviewed here are suggestive, but far from conclusive. Nonetheless, they suggest that the group theoretic perspective presented can provide a new way to think about non-linear response properties in visual cortex. In the remainder of the thesis, I demonstrate how group representations and the generalized bispectrum can be used to construct artificial neural networks with improved selectivity, robustness, and invariance.

Chapter 4

Learning Groups and Their Representations in Neural Networks

Natural data contain complex, hierarchical transformation structure that is difficult to specify *a priori*. For example, in a visual scene, translations, scalings, rotations, diffeomorphisms, and lighting changes all interact in complex ways on multiple scales—on parts, objects, and whole scenes. In addition, this inherently 3D structure is lossily collapsed into an underdetermined 2D image when a measurement is taken. This makes representing group structure in natural signals a problem of *inference* and *estimation*.

In this chapter, I apply the ideas established in Chapter 3 to the design of artificial neural networks with improved robustness and selectivity in the context of *learning unknown transformation structure*. I propose a neural network model for learning robust, selective invariance to group structure in data, inspired by the analytical generalized group-invariant bispectrum. The model builds in very minimal structure—only the general functional form of the bispectrum as a third-order, complex valued-map, with no group representations built in. The rest of the map is parameterized and left to optimization.

In two experiments, across both commutative and non-commutative groups, the model learns robust group invariant maps. Moreover, the weights learned in the network approximate the irreducible representations for the group underlying the transformation structure in the datasets, and the full map approximates the analytical group-invariant bispectrum. The model is compared to generic multi-layer ReLU networks, which require up to 100x as many parameters to achieve similar performance, and demonstrate less robustness and selectivity.

4.1 The Problem: Learning a Complete Invariant Map

To learn robust mappings that respect the transformation structure in natural data, a model must possess the mathematical capacity to efficiently parameterize such maps. Currently, much of generic deep learning is built on generic point-wise non-linearities and pooling operations that are highly non-selective, and fail to ensure the robust *selective invariance* that biological vision achieves.

We take a top-down approach, starting from a known solution to the problem of complete invariance: the *bispectrum*, an analytical object that is a complete invariant for arbitrary compact groups. We are interested in *learning* bispectra for arbitrary groups, without *a priori* knowledge of the group structure in the data.

To that end, we present a novel neural network model —Bispectral Neural Networks (BNNs) —that parameterizes the general form of the bispectrum. We train these networks in both supervised and semi-supervised contexts to either explicitly generate a bispectrum (Section 4.3) or learn to collapse group orbits (Section 4.4). Across both contexts, the parameters learned in the model approximate the irreducible representations of the groups in the data. As a consequence, the linear sub-components of the BNNs are *equivariant* to the group actions, which allows the network to simultaneously characterize relative distance between points in the same orbit. Our shallow model is suitable for hierarchical stacking in deep learning. Here, we demonstrate the properties of a single bispectral layer for simultaneously learning invariant and equivariant maps.

4.2 Model: Bispectral Neural Networks

We draw inspiration from the generalized bispectrum to define Bispectral Neural Networks, a neural network module that parameterizes the *ansatz* of the bispectrum. The motivating idea behind this approach is to constrain the optimization problem by building in the general *functional form* of a solution known to exist. Importantly, we build in only the generic functional form and allow the network to learn the appropriate parameterization from data.

We define two variants of a Bispectral Network architecture, depicted in Figure 4.1, with different amounts of mathematical structure built in. The first—the *Full Bispectral Network*—has a single complex-valued weight matrix $W \in \mathbb{C}^{d \times n}$ that defines a linear transform on the data $x \in \mathbb{R}^d$, with d = the data dimension and n = the number of neurons. The output of the module is computed as all triple products

defined by the following indexing relationship:

$$B_F(x)_{k_1, k_2} = Wx_{k_1} Wx_{k_2} Wx_{k_1+k_2}^\dagger, \quad (4.1)$$

with the summation $k_1 + k_2$ taken modulo the dimension d . This yields an output matrix of dimension $n \times n$, which is raveled into an n^2 -dimensional vector. If W is a Discrete Fourier Transform (DFT) matrix, the output of the module will be exactly equal to the translation-invariant bispectrum. Thus, if through the training procedure the architecture learns a bispectrum-related solution, the weight matrix W should approximate an ordered DFT matrix.

The full bispectrum has n^2 terms (with n = number of frequencies). However, it is highly redundant and can be reduced according to its 12-fold symmetry without loss of information (See Chapter 3, Figure 3.2). Thus, we hypothesize that it can be approximated in a computationally simpler form that does not require taking all pair-wise triple products. We propose a second architecture—the *Approximate Bispectral Network*—that exchanges computational complexity for some additional memory complexity. Concretely, the Approximate Bispectral Network has three weight matrices, U , V , and $W \in \mathbb{C}^{m \times n}$, each of which defines a linear transform on the data. The output of these three terms are combined in a Hadamard (element-wise) product:

$$B_A(x) = Ux \odot Vx \odot Wx \quad (4.2)$$

No additional non-linearity is used beyond the triple product. If, as predicted, the architecture learns a bispectrum-related solution, then U , V , and W will tend towards the irreducible representations of the group acting on the data. In this case, each of the three linear maps (e.g. Ux) will approximate a generalized Fourier transform and should have all of the associated properties, including equivariance to group action in the input.

4.3 Experiment 1: Supervised Learning of the Bispectrum

Experiment 1 first tackles the problem of learning the bispectrum with supervision—i.e. learning the map from data vectors x to their translation-invariant bispectra $\beta(x)$ from $(x, \beta(x))$ pairs. In this context, the bispectrum of each datapoint is

Bispectral Network Architectures

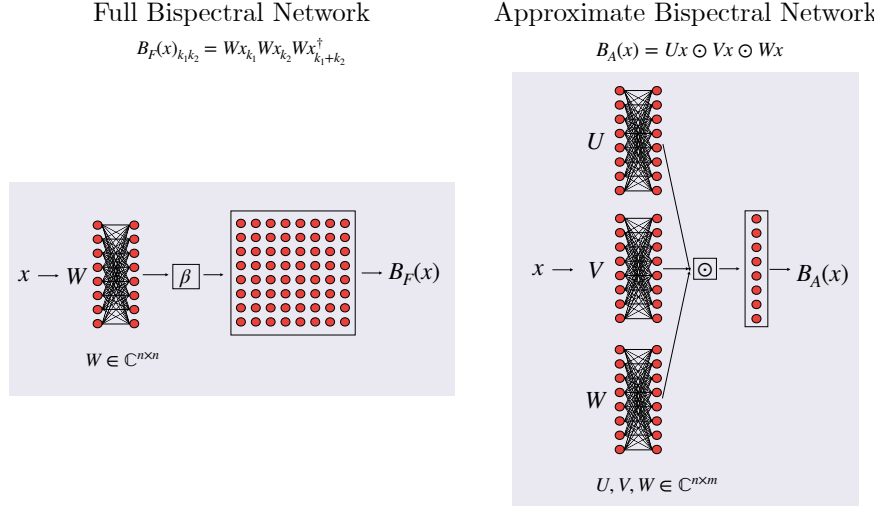


Figure 4.1: The Full and Approximate Bispectral Network Architectures. The Full Bispectral Network computes all pairwise triple products defined by the indexing relationship $(k_1, k_2, k_1 + k_2)$ and outputs a matrix of dimension $m \times m$. The Approximate Bispectral Network reduces the computational complexity of the Full Network by introducing three weight matrices, U , V , and W , the outputs of which are element-wise multiplied. The output of the Approximate Network is a vector of dimension n .

its “label,” and the network is trained to learn to output this label by minimizing the mean squared error between the network output and the ground truth bispectrum.

Concretely, data samples are represented as functions $x^i : M \rightarrow \mathbb{R}^d$ sampled at d regular intervals over an input manifold M . Given a set of c “canonical” datapoints $\{x^1, \dots, x^c\}$ defined on the manifold M , and a group G that acts on M with group actions α_g , the orbit of each datapoint is defined $\{x^i(\alpha_g m) | g \in G\}$. The dataset consists of the union of the orbits of the n canonical datapoints $X = \bigcup_{i=1}^c \alpha_G x^i(M)$, with each orbit defining an equivalence class, with each datapoints labeled by its bispectrum $\beta(x^i)$. The goal is to determine whether the proposed network architectures or a generic multi-layer neural network can learn to approximate the bispectrum map from examples.

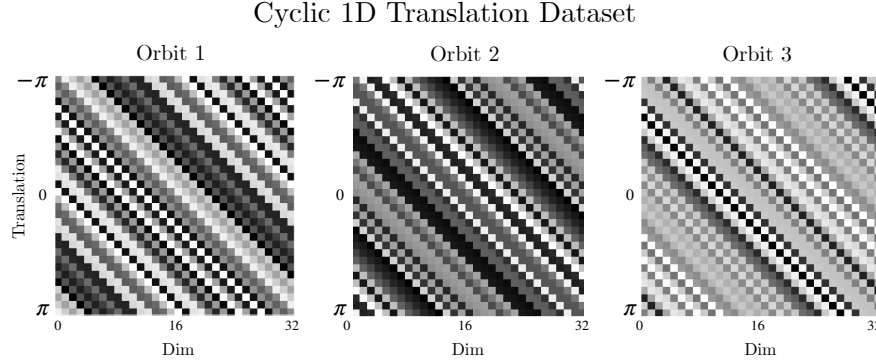


Figure 4.2: Cyclic 1D Translation Dataset. Data vectors in 3 example orbits are shown. Each row is a data vector, which is ordered by its cyclic translation on the y-axis.

4.3.1 Methods

We generate 100 random data vectors $x \in \mathbb{R}^{32}$, with I.I.D. components drawn from a normal distribution $N(0, 1)$. We generate the orbits of each datapoint under the group of cyclic 1D translations, yielding 32 single-pixel translations of each canonical datapoint. The bispectrum of each orbit element is computed and stored as its “label.” The set of bispectral labels were standardized to zero mean and unit standard deviation. Note that the bispectra of each within-orbit datapoints will be equivalent. Thus, by learning the map between datapoints and their bispectra, the network will implicitly learn invariance to translation. Moreover, because the bispectrum is complete, a successful network would learn a robust, selective invariant. A random 20% of the dataset is set aside for model validation and is used to tune hyperparameters. The remaining 80% is used for training.

We compare the two Bispectral Network architectures depicted in Figure 4.1 to a generic multi-layer ReLU Network architecture, with the number of layers and neurons in the ReLU Networks varied as hyperparameters. Bispectral Networks have weights in $\mathbb{C}^n$ and use no additional non-linearity beyond the third-order product that defines the bispectral layer. ReLU Networks have weights in $\mathbb{R}^n$ and use ReLU activations between all layers except for the last.

Bispectral Networks were initialized using the complex orthogonal weight initialization method proposed in [131], which we call Trabelsi Complex Orthogonal here. ReLU Networks were initialized with the Kaiming uniform method [45]. All networks were implemented and trained in PyTorch [98]. We additionally made use of the open-source CplxModule library [90], which provides PyTorch implementations of the various complex-valued operations and initializations used in this work. We used

a batch size of 32 for all networks. In addition, for all networks, the learning rate was reduced by a factor of 0.1 whenever the validation loss plateaued for 10 epochs. This common learning rate scheduling strategy helps the network to escape local minima within saddle-points in a loss landscape. All networks were trained using the Adam optimization method [67] using a learning rate of .001.

Learning Curves for Supervised Learning of Bispectra

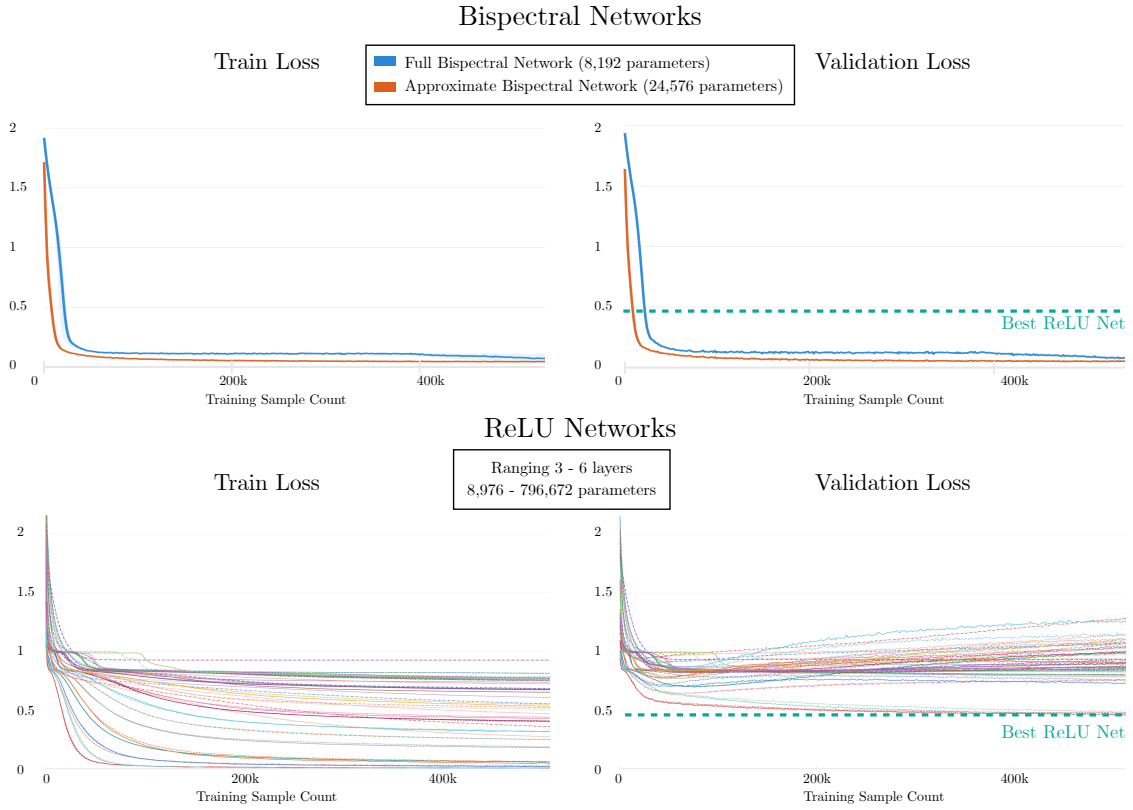


Figure 4.3: Learning Curves for Supervised Learning of Bispectra. Loss performance on the training (Left) and validation (right) datasets are depicted for Bispectral Networks (Top) and ReLU Networks (Bottom). ReLU plots show curves for 26 architecture variants, ranging from 3 - 6 layers and 8,976 - 796,672 parameters. Depicted Bispectral Networks used 8,192 (full) and 24,576 (approximate) parameters.

4.3.2 Results

Training and Validation Performance

Figure 4.3 shows learning curves for ReLU and Bispectral Networks on this task. The ReLU figures show curves for 26 architecture variants, ranging from 3 - 6 layers and 8,976 - 796,672 parameters. All of the ReLU Networks (bottom) struggled to learn the map. While many of the ReLU Network architectures successfully achieve a low MSE reconstruction on the training set, none of the networks achieve a loss lower than 0.44 on the validation set. This best-performing network had a 5-layer architecture of the form 256-256-256-256-512 (4 layers with 256 neurons plus an output layer with 512), and used a bias term in each layer (models tested without bias performed comparably), for a total of 333,540 parameters.

By contrast, both Full and Approximate Bispectral Networks (top) quickly learn to approximate the analytical bispectrum, achieving validation loss of 0.04 and 0.03, respectively. The networks require only 8,192 (full) and 24,576 (approximate) parameters —counting the real and imaginary components of the weights as separate parameters. Remarkably, these relatively small networks vastly outperformed the best ReLU Network, which required 40x (Full) or 13.5x (Approximate) the number of parameters and performed considerably worse. While further increases in the number of layers or parameters may increase the performance of the ReLU Networks, the strong performance and high efficiency of the Bispectral Networks at this task makes a strong case for their use over an even larger ReLU Network.

Learned Weight Structure

An examination of the weights in the Full and Approximate Bispectral Networks confirms that the map learned by the networks approximate the analytical bispectrum map. Figures 4.4 and 4.5 show the weights learned in the Full and Approximate networks, respectively. Each weight vector in the learned Full Bispectral Network is selective for a different frequency, as revealed by the power spectrum of each vector (right). Moreover, the network learns to order them to satisfy the indexing constraint imposed by the architecture. Thus, the network learns to approximate an ordered DFT matrix. It should be noted that no constraint on the norm of the weight vectors was imposed. Consequently, the norm of each vector ranges and is not typically unit length.

In the Approximate Bispectral Network, three weight matrices (U, V, and W) are learned. We expect that the network should learn to pair frequencies element-wise across each matrix such that the frequency of one weight vector is equal to the negative

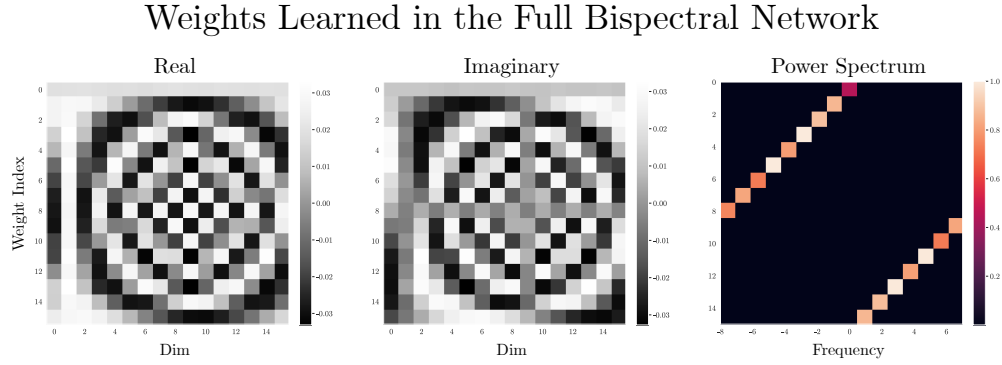


Figure 4.4: Weights Learned in the Full Bispectral Network. The weight matrix learned in the Full Bispectral Network approximates an ordered Discrete 1D Fourier Transform Matrix, with each weight vector (row) approximating a different 1D Fourier basis, as revealed by the power spectrum of each vector.

of the sum of the frequencies of the other two weight vectors. It should be noted that the ordering of this relationship will be permuted for each triplet of weights—that is, the frequency of the first weight in U might be the negative sum of the frequencies of the first weights in V and W , whereas the second weight in W may play the negative sum role.

Figure 4.5 (top) shows the weights learned in the Approximate Bispectral Network. Nearly all learned weight vectors (left) approximates a single 1D Fourier basis function, as revealed in the power spectrum (right). However, a small subset of the learned weights were “null” and essentially pushed to zero. Here, only the real components out of the first 64 out of 256 weight vectors are shown. In addition, the network learns the appropriate frequency relationships across the triplets to compute analytical bispectral coefficients.

To quantify this, each weight vector is identified with the maximum power frequency in its Fourier transform. Triplets of weight vectors are analyzed to determine whether the bispectral frequency relationship holds: $k = -(i + j)$, for some permutation of i, j , and k , with the sum modulo d . This analysis reveals that 73.44% of the triplets in the approximate network are bispectral triplets, with the remaining triplets involving at least one of the “null” non-frequency-selective weight vectors. The learned first-layer weight matrices in the best ReLU net (Figure 4.5, bottom) show considerably less Fourier structure. However, the power spectrum reveals some frequency selectivity in the weights.

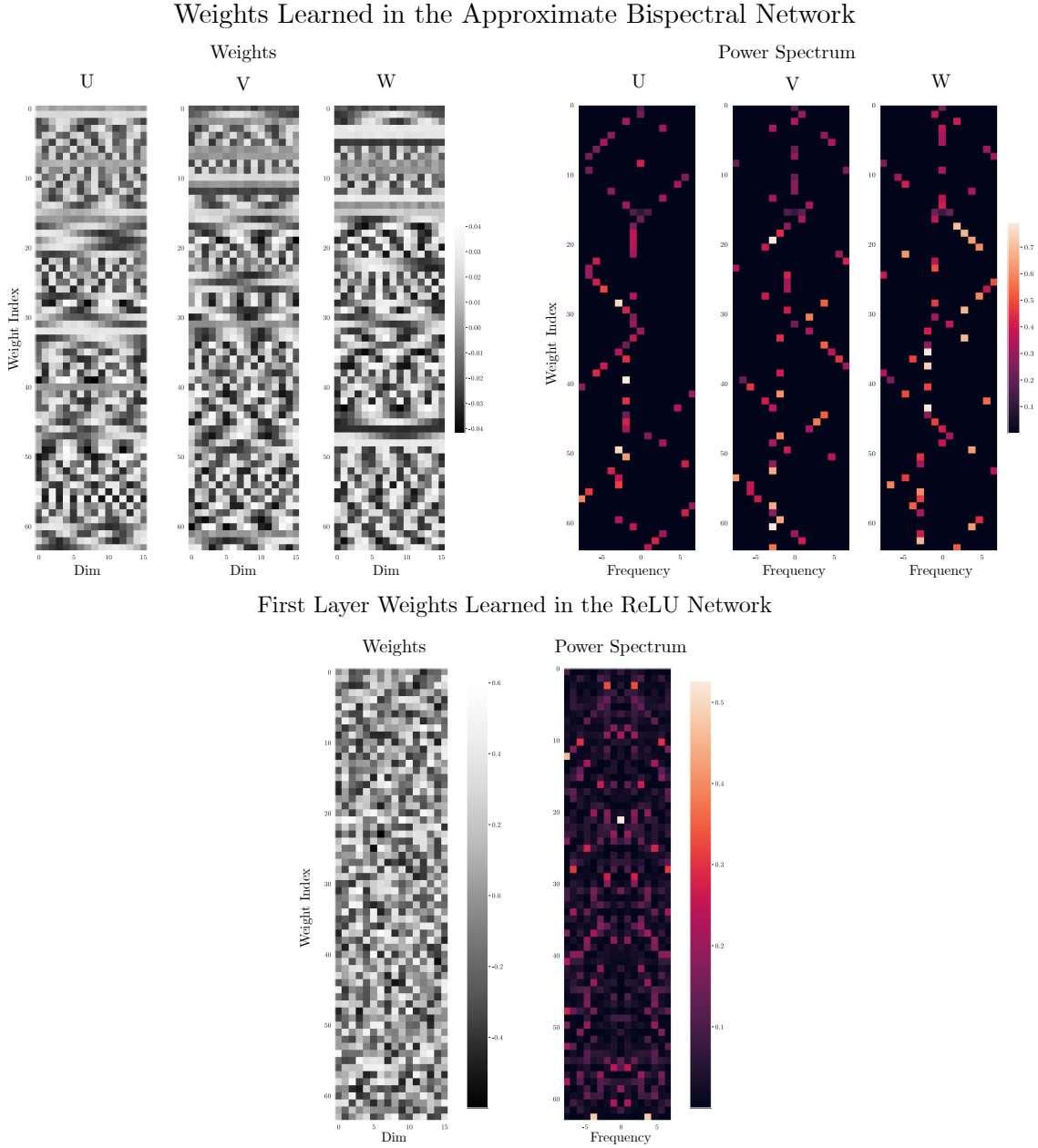


Figure 4.5: First-Layer Weights Learned in the Approximate Bispectral Network and ReLU Network. (Top) The first 64 vectors in the U, V, and W weight matrices learned in an Approximate Bispectral Network on the supervised bispectrum learning task. Only the real components are displayed here. (Left) Learned weight vectors approximate 1D Fourier basis functions. (Right) The power spectra of the learned weight vectors. The spectrum of each weight vector is a delta function centered at a single frequency. (Bottom) The first 64 weight vectors in the first-layer weight matrix of the ReLU Network. Weight vectors show some selectivity for 1D Fourier frequencies, as illustrated in the power spectra of each weight vector, but the power is typically distributed over multiple frequencies.

Generalization to Novel Data

If the networks have successfully learned the bispectrum map, they should be able to approximately compute bispectra for *any* arbitrary novel data vector. The outputs of each network were evaluated on a test dataset consisting of 100 novel vectors generated from the same distribution as the training dataset and their orbits under cyclic translation, for a total of 1600 test datapoints. Outputs of the networks were compared to analytically computed bispectra and evaluated according to the mean Euclidean distance to the analytical solutions. Table 4.1 shows test performance for the best of the three network architectures. Both the Full and Approximate Bispectral Network robustly generalize to entirely novel data, computing bispectral approximations that are within 0.033 and 0.027 of the analytical bispectra. The ReLU Network, by contrast, failed to generalize, resulting in a mean squared error of 1.052, approximately equal to the standard deviation of the bispectral labels themselves.

Architecture	Test Loss	Parameters
ReLU	1.052	333,540
Approximate Bispectral	0.027	24,576
Full Bispectral	0.033	8,192

Table 4.1: Generalization Performance for Supervised S1. The Approximate and Full Bispectral Networks generalize robustly to unseen data, approximating the analytical bispectra of these datapoints. The ReLU Network, by contrast, fails to generalize.

4.3.3 Discussion

Experiment 1 establishes that a single-layer complex-valued neural network with a third-order element-wise product (Bispectral Network) can learn the bispectrum map in a supervised training regime, in which datapoint examples and their corresponding bispectra are presented. It also demonstrates that a generic multi-layer neural network architecture using real-valued weights and rectified pointwise non-linearity after each layer (ReLU Network) struggles to learn this map in the same setting, even when permitted 40x as many parameters as the Full Bispectral Network. The results of this experiment emphasize that the choice of computational primitives is critical. While a ReLU Network can in principle approximate any continuous function on a compact domain, it may in practice require exponentially many neurons to do so [22, 37, 52, 6]. Building in the correct ansatz to solve this canonical and complete solution to group invariance allows neural networks to efficiently parameterize these solutions in an interpretable manner.

The primary goals of this experiment were to determine (1) the general feasibility of

training our proposed network architecture, (2) whether the network could learn to represent the bispectrum in a supervised setting, providing maximal information, (3) whether the Approximate Bispectral Network would learn to approximate the analytical bispectrum, and (4) the advantages or disadvantages, if any, of these architectures as compared to a generic multi-layer neural network on the same task.

Having established (1), (2), and (3) in the affirmative, as well as the superiority of this architecture as compared to a generic multi-layer ReLU Network (4), we now consider a more naturalistic, semi-supervised learning setting in which the group structure that defines orbits in the data is unknown, with the goal of learning generalized bispectra for arbitrary groups beyond translation. Importantly, when we move to groups beyond translation, we will need to use the Approximate architecture only, as the indexing relationship $(i, j, i + j)$ will not necessarily hold for the irreducible representations of arbitrary groups. Therefore, for all of the studies under Experiment 2, we consider only the Approximate Bispectral Network architecture.

4.4 Experiment 2: Learning Group Invariant Maps

We now consider the problem of learning invariance to an *unknown* group acting on a dataset and pose it as a *representation learning* or *embedding* problem. Given a dataset with latent group structure and labels indicating which datapoints are equivalent up to transformation (but without information about the transformation itself, i.e. *how* they are related), we wish to learn an invariant map that collapses elements of the same group orbit and separates distinct orbits. Critically, we want this map to be both *invariant* to group actions and *selective* to pattern structure—that is, to be *complete* [63]. A network that successfully learns such a map will have implicitly *learned the group* that structures transformations in the data.

In [44], Hadsell, et al. proposed a loss function to solve a similar problem. The motivation behind the *contrastive loss* is to learn a map from datapoints to an embedding space in which datapoints considered similar are nearby and datapoints considered dissimilar are distant. Given a set of data vectors $X = \{x_1, \dots, x_n\}$ with $x_i \in \mathbb{R}^d$, a neural network $f^\theta : X \rightarrow Y$ with parameters θ , and output vectors $Y = \{y_1, \dots, y_n\}$ with $y_i \in \mathbb{R}^{d'}$, define the binary labeling function:

$$Y(x_i, x_j) = \begin{cases} 0 & \text{if } x_i \text{ and } x_j \text{ are similar} \\ 1 & \text{otherwise} \end{cases}$$

The contrastive loss is defined as:

$$L(y_i, y_j) = \frac{1}{2} \left(1 - Y(x_i, x_j) \right) D(y_i, y_j)^2 + \frac{1}{2} \left(Y(x_i, x_j) \right) \max(0, m - D(y_i, y_j)^2) \quad (4.3)$$

where y_i and y_j are pairs of network outputs, i.e. “embeddings”, D is a distance function such as the Euclidean distance, and $m > 0$ defines the minimum permitted margin of distance between similar and dissimilar datapoints. The binary labeling function Y ensures that different penalties are calculated for similar and dissimilar pairs. If the pair of points are similar, only the first term will contribute to the loss, and the penalty will be proportional to the distance of the embeddings. If the pair of points are dissimilar, only the second term will contribute to the loss, and only if the distance of the two embeddings is within the margin m , with a penalty proportional to how close together the embeddings are. That is, if two dissimilar points are close in embedding space, the loss function will encourage them to farther apart. The classification of points as similar is specified by the engineer, and can be chosen or calculated using prior knowledge—for example, the assumption that temporally proximal video frames are visually similar.

We pose a group theoretic framing of the contrastive loss to apply it to the problem of collapsing group orbits. Datapoints are represented as functions $x_i : M \rightarrow \mathbb{R}^d$ sampled at d regular intervals over an input manifold M . Given a set of datapoints $X^M = \{x_1, \dots, x_n\}$ defined on the manifold M , and a group G that acts on M with group actions α_g , we define the labeling function:

$$Y(x_i, x_j) = \begin{cases} 0 & \text{if } \exists g \in G \mid x_j(\alpha_g(M)) = x_i \\ 1 & \text{otherwise} \end{cases}$$

Under this framing, with labels defining group orbits, the contrastive loss of Equation 4.3 approximately captures the notion of *complete group invariance*, as it pushes datapoints that are equivalent up to group action to the same point and datapoints non-equivalent up to group action to different points.

The bispectrum is one known complete solution to the orbit separation problem. Here we ask the following:

1. Will a neural network trained to minimize 4.3 on a dataset with unknown group structure learn to approximate the generalized bispectrum for that group?

2. Is a Bispectral Network more likely to find this solution than a generic multi-layer ReLU Network?
3. If so, are the learned approximations of this solution more *complete* than the solutions found in generic architectures?

4.4.1 Methods

In addition to the 1D cyclic translation dataset used in Experiment 1, the networks are trained and tested on synthetic datasets for the groups of 2D rotation, 2D scaling, and 3D spherical rotation. The structure of these datasets is defined by the *space* over which the data is defined, the *group* G that acts on the data space to define the orbits of data samples, and the *pattern* statistics that generates samples. The datasets are summarized in Table 4.2.

Space	Group	Transformation	Pattern
S^1 Ring	$\mathbb{Z}/n\mathbb{Z}$	1D Cyclic Translation	$x_i^d \sim \mathcal{N}(0, 1)$
$\mathbb{R}^2$	$SO(2)$	2D Rotation	T^2 Harmonics
S^2 Sphere	$SO(3)$	3D Rotation	S^2 Harmonics

Table 4.2: The Synthetic Datasets Used in Experiment 2.

As in Experiment 1, datapoints are represented as functions $x : M \rightarrow \mathbb{R}^d$ sampled at d regular intervals over a manifold M . The space $S^1 = \{g \in \mathbb{C} : \|g\| = 1\}$ is used to define time-varying, real-valued signals and their cyclic translation in time via a group action of multiplication by a unit vector in $\mathbb{C}$. The space $\mathbb{R}^2$ is used to represent in-plane rotations of images using the action of the *special orthogonal group* $SO(2)$ of pure rotation matrices. For the group of 3D spherical rotations $SO(3)$, signals are defined over the sphere S^2 , represented in spherical coordinates.

The group of cyclic 1D translations can be naturally discretized as the group $\mathbb{Z}/n\mathbb{Z}$, the group of integers modulo n , with n the dimension of the signal. Thus, data vectors with components drawn from an i.i.d. normal distribution (i.e. the dataset used in Experiment 1) will not be distorted when cyclically translated in integer step sizes. For the continuous rotation and scaling groups, we must use an alternative pattern generating process, to avoid distortions when the patterns are rotated and scaled on discretized manifolds. For these datasets, patterns are generated as linear combinations of a random subset of continuous harmonic functions, with coefficients drawn from a uniform distribution. This generates patterns with continuous structure, thus reducing discretization errors. For the datasets defined on the plane $\mathbb{R}^2$, patterns consist of linear combinations of plane waves, i.e. the harmonics of the 2-Torus T^2 .

For the $SO(3)$ dataset of spherical rotations, patterns consist of linear combinations of spherical harmonics. To test the generalization ability of the network trained for $SO(2)$ rotations, a synthetically rotated version of the Omniglot dataset [73] is used. The specific parameters used to generate each dataset are described in each study below.

As in Experiment 1, all models are implemented and trained in PyTorch [98] and Cplx-Module [90]. The Trabelsi complex orthogonal weight initialization method [131] is used for the Bispectral Network, and the Kaiming Uniform method [45] for the ReLU Networks. For the calculation of the contrastive loss and associated sampling schemes, the PyTorch Metric Learning [89] implementations are used. On each training step, mini-batches are selected use the m-per-class sampling strategy, which ensures that each batch contains m samples of as many classes will evenly divide the batch size—i.e. for $m = 10$ and a batch size of 50, each batch will contain 10 samples of 5 classes. This strategy ensures that each batch will have sufficient positive and negative samples to compute the two terms in the loss function on every gradient update step. All models are trained with the Adam optimizer [67].

Each trained model is evaluated on its *training and validation performance, learned weight structure, generalization performance, and the properties of the embedding space under transformation.*

4.4.2 Cyclic Translation S_1

The networks are first trained on the group that defines the canonical 1D Fourier transform and bispectrum: the group of one-dimensional cyclic translations. The goals of this study are to determine (1) whether neural networks trained to collapse group orbits with the contrastive orbit loss will succeed at this objective, (2) whether the learned map will be robust, selective, and will generalize to novel data and (3) whether the learned first-layer and full-network maps will approximate the translation-invariant Fourier transform and bispectrum, respectively. Starting with a simple, well-understood case facilitates the analysis of the learned network structure using the canonical Fourier transform and bispectrum.

As in Experiment 1, the Bispectral Network is compared to a generic multi-layer ReLU Network. If the Bispectral Network learns a bispectrum-like solution, the weights in the network should approximate the 1D Fourier basis and the bispectral indexing relationship should hold between the triplets of frequencies in the U , V , and W matrices. It will be more difficult to validate whether the ReLU Network learns to approximate a bispectrum, as the structure will be less explicit. If 1D Fourier filters emerge in the first-layer weights, this suggests a Fourier-based solution, but

is ambiguous between the power spectrum, bispectrum, or an alternative strategy. To disentangle power spectral and bispectral solutions, the networks are tested on an “adversarial” dataset consisting of “adversarial” orbits, in which data vectors have the same power spectra, but are not equivalent up to translation, and thus have distinct bispectra. A network that approximates a power spectral solution should collapse these datapoints to the same point in embedding space, whereas a network that learns a bispectral solution will not.

Methods

The training dataset is generated using the same procedure as in Experiment 1. The dataset consists of 100 random vectors $x \in \mathbb{R}^{32}$, with i.i.d components $x^d \sim \mathcal{N}(0, 1)$. The orbit of each of these patterns is generated as all cyclic translations of the pattern, in increments of 1 pixel. A random 20% of the dataset is set aside for model validation and is used to tune hyperparameters. The remaining 80% is used for training. Figure 4.2 shows example datapoints.

A primary objective is to test whether the models will generalize robustly to novel, unseen data. The test dataset consists of 100 new random vectors drawn from the same distribution as the training dataset. In addition, generalization performance of the networks under noise is examined on noisy versions of the test dataset, in which Gaussian noise is added to each component of each data vector. Three noisy test datasets are generated with increasing levels of noise drawn from $\mathcal{N}(0, 0.1)$, $\mathcal{N}(0, 0.2)$, and $\mathcal{N}(0, 0.3)$. Figure 4.11 shows examples from the noisy datasets.

The adversarial test dataset consists of phase-scrambled “adversarial” orbits that have equivalent power spectra but non-equivalent bispectra. The dataset is generated from a new set of 100 random vectors $x \in \mathbb{R}^{32}$, with i.i.d components $x^d \sim \mathcal{N}(0, 1)$. As in the training dataset, all cyclic translations of each vector are generated. The power spectra of each orbit is computed and the magnitude spectra is preserved. Random phase spectra for each data vector are generated from $\mathcal{U}(-\pi, \pi)$, and are then combined with the magnitude spectra and pushed through the inverse Fourier transform to generate adversarial patterns in pixel space.

As in Experiment 1, Bispectral Networks are compared to generic multi-layer ReLU Networks. The number of layers (between 3 and 6) and number of neurons in each layer (between 32 and 512) are varied as hyperparameters to determine the best performing ReLU architecture. No non-linearity is used in the output layer of the ReLU networks. A hybrid model, which consists of a Bispectral layer followed by a single linear layer with no output non-linearity (2-Layer Bispectral Network), is additionally tested to examine the effect of an additional linear layer on model performance. All

networks use the same weight initialization and optimization method from Experiment 1, and learning rates of 0.001.

Results

Training and Validation Performance

The training and validation performance for the best of each of the three architectures is displayed in Table 4.3 below. All three architectures converge to a low training and validation loss, with the 2-Layer Bispectral Network achieving the best validation performance. The best performing ReLU Network had 3 layers of 512 neurons each, with bias terms, for a total of 804,352 parameters. The 1- and 2-Layer Networks used an efficient 6,144 and 8,192 parameters, respectively. A comparably sized ReLU Network, with six layers of 32 neurons each (7,168 total parameters) achieved training loss of .7412 and validation loss of .8728. This underscores that the ReLU Network needs many more parameters to achieve comparable performance on this task—98x as many, when compared to the 2-Layer Bispectral Network.

Architecture	Train Loss	Validation Loss	Parameters
ReLU	0.0004	0.0214	804,352
Bispectral 1-Layer	0.0826	0.0763	6,144
Bispectral 2-Layer	0.0003	0.0003	8,192

Table 4.3: Training and Validation Performance on Cyclic 1D Translation for the best of each architecture tested. The real and imaginary components of each neuron are counted as separate parameters for the Bispectral Networks.

Weight Structure

Figure 4.6 shows the learned weights in the best-performing Bispectral Networks (1-Layer, left; 2-Layer, middle) and ReLU Network (right). Remarkably, across all architectures, first-layer weights learn to approximate the 1D Fourier basis. This is explicitly revealed in a power spectral analysis of the weight matrices (Figure 4.7). While the Fourier component structure is implicitly contained in the bispectrum labels provided in Experiment 1, this experiment involves no Fourier components at all; the networks observe only translating versions of random data vectors and group orbit labels. It is notable that the ReLU network learns the Fourier basis in this setting, as it failed to do so in Experiment 1. This suggests that the more relaxed contrastive orbit objective allows the architecture to find alternative (potentially non-bispectral) solutions. Despite the relaxation, the solution it finds appears to be Fourier-based.

First-Layer Weights in Bispectral and ReLU Networks

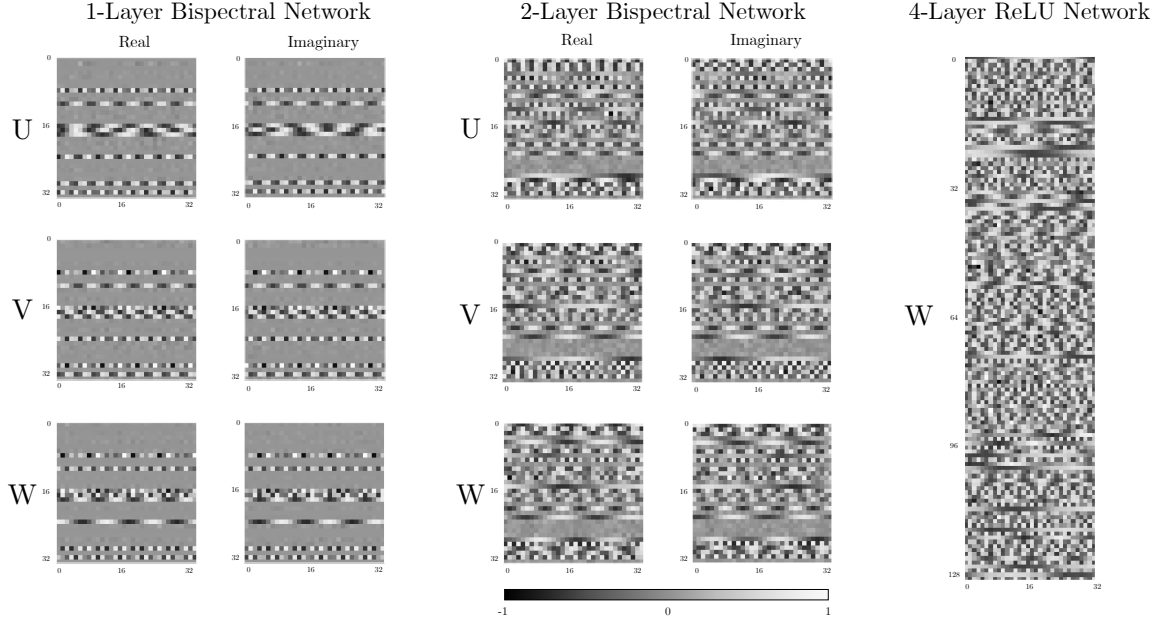


Figure 4.6: First-Layer Weights in Bispectral and ReLU Networks. Learned first-layer weights in 1-Layer (Left) and 2-Layer Bispectral Networks (Middle), and a 4-Layer ReLU Network (Right). Each row in each matrix is a weight vector of dimension 32. Real and imaginary components are depicted separately for the Bispectral Networks. All weights have been normalized to have a minimum value of -1 and maximum value of 1.

It is also notable that the 1-Layer Bispectral Network learns only approximately 1/3 of the Fourier basis, and pushes the remaining weights towards zero. The 2-Layer Bispectral Network learns a larger subset of the 1D Fourier basis. However, both bispectral architectures contain some proportion of “null” weights (e.g. weight indices 24 - 27 in the 2-layer network, Figure 4.6). This pattern holds across random seeds and random datasets. The ReLU Network, by contrast, learns no null weights, with each learned weight vector approximating a 1D Fourier basis function. This suggests that the third-order complex-valued structure of the bispectral layer may introduce complexity into the optimization process in this setting.

To examine the frequency relationships between the triplets of weights that are multiplicatively combined in the Bispectral Network, each weight vector is identified with its max power frequency. Figure 4.7 displays the triplets of frequencies for the first-layer weights learned in the 2-Layer Bispectral Network. Of the 32 weight vector triplets, 71.9% of the triplets possess the frequency relationship defined by the ana-

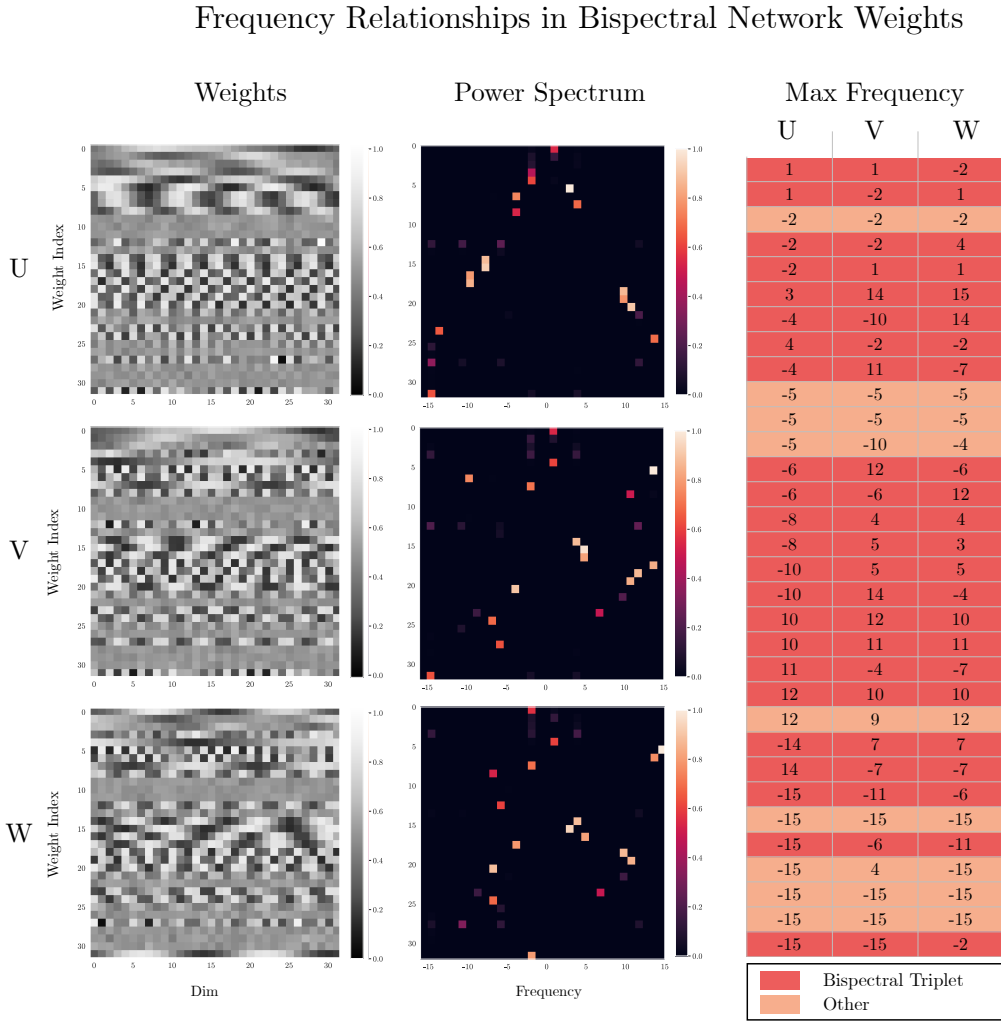


Figure 4.7: Frequency Relationships in the Bispectral Network Weights. Learned weights (Left) and their power spectra (Middle) in the Bispectral Network reveal general selectivity for 1D Fourier bases. An examination of the frequency selectivity for the triplets of neurons in U , V , and W that are multiplicatively combined reveals an approximate bispectral strategy: 71.9% of these triplets possess the bispectral frequency relationship (Right).

lytical bispectrum. The remaining triplets include null weights. The robust prevalence of this triplet structure suggests a successful approximation of the analytical bispectrum.

Map Structure

We now examine the equivariance and invariance properties of the transform itself. Figure 4.8 depicts the outputs of the first linear term and full network output for both the Bispectral Network (here: the 2-layer network) and the ReLU Network for three example orbits. On the x-axis of each plot, the translation of the datapoint is continuously varied. The output value corresponding to each neuron in the network is depicted by a different colored line, as the patterns are translated. The y-axis depicts the resulting activation of each neuron. The outputs of three select neurons are displayed for the first linear term, for clarity.

The output of each linear term in the first layer of the Bispectral Network is perfectly equivariant to translations in the input, with the phase shift resulting from a single pixel translation proportional to the frequency of each neuron—as expected of an analytical Fourier transform. This can be replicated by comparing to the curve $e^{-i\pi\theta x}$, with θ the preferred frequency of the neuron and x ranging from 1 to the signal dimension n . The output of the full network is perfectly invariant to translations in the input—as expected of an analytical bispectrum. The output of the first layer of the ReLU Network exhibits rectified equivariance, with some noise. The linear transform defined by the ReLU network’s learned Fourier component weights is equivariant to translation. However, the rectification of the first layer truncates all negative values to zero, thus introducing discontinuities. The full output of the ReLU Network is largely invariant to translation, with some noise.

Generalization to Novel Data

Model performance is examined on the test dataset, noisy test datasets, and the adversarial dataset. Model outputs for each dataset are computed. Model performance is quantified by selecting the k -nearest-neighbors of each datapoint and calculating the proportion that are within-orbit, with k equal to the size of the orbit and the Euclidean distance determining nearest neighbors. A model that has perfectly and selectively collapsed orbits will show 100 % within-orbit nearest-neighbors for all datapoints. The within-orbit nearest-neighbor percentage is averaged across test datapoints.

Figure 4.9 compares the performance of the best Bispectral and ReLU Networks on the generic test dataset. For visualization purposes, Euclidean distance matrices are displayed for the first 10 orbits in the test dataset. Remarkably, both networks generalize robustly to the novel dataset, demonstrating clear and robust invariance to translation with near-zero distance for all within-orbit datapoints and perfect (100%) or near-perfect (99.78%) orbit identification for the Bispectral and ReLU Networks,

Model Outputs on Translating Data

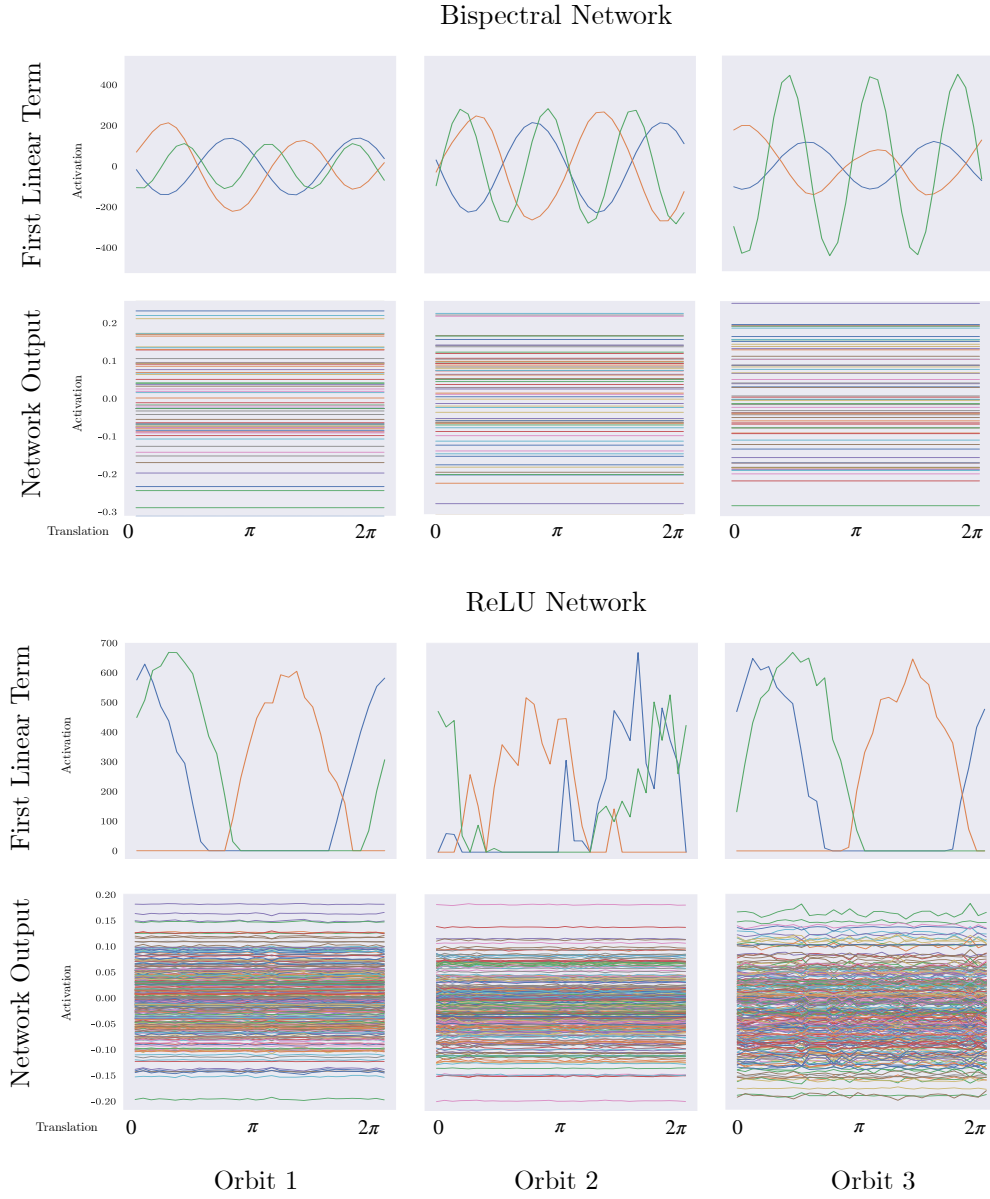


Figure 4.8: Model Outputs on Translating Data. Outputs of the first linear term and full network output are displayed for the Bispectral Network (Top) and ReLU Network (Bottom). Data inputs are cyclically translated (x-axis) and layer/network outputs are computed for each translation. Each colored line represents the output of a single neuron, with the value of the activation on the y-axis. For the first linear term, only three neuron outputs are visualized, for clarity.

respectively. This is particularly notable for the ReLU Network, which was unable to learn the analytical bispectrum map in the supervised context of Experiment 1. This suggests that the reduced constraints of the contrastive loss experiment have permitted the ReLU Network to find an alternative solution. We now examine the properties of these solutions with the adversarial and noisy datasets.

The ReLU Network’s responses on the phase scrambled “adversarial” dataset are revealing of the underlying structure of the learned map. Figure 4.10 shows pairwise distance matrices and k -nearest-neighbor analyses for Bispectral and ReLU Network outputs on the adversarial dataset. In this phase-scrambled data, the datapoints are non-equivalent up to translation and thus have distinct bispectra. However, datapoints within the same “adversarial orbit” have the same power spectra and would thus be collapsed to the same point by a power spectrum map. The distance matrix for the ReLU Network on phase-scrambled data shows clear (though imperfect) block diagonal structure, indicating that the network has partially collapsed these adversarial orbits.

The Bispectral Network, by contrast, shows no block-diagonal similarity structure. This is demonstrated quantitatively by the k -nearest-neighbor analysis, which demonstrates that the Bispectral Network collapses 3.66% of the datapoints within the adversarial orbits on average, while the ReLU Network collapses an average of 12.42% of the items within adversarial orbits. Notably, the Bispectral Network is close to chance levels at collapsing adversarial orbits, as the expected percentage would be 3.12% if orbits were assigned randomly. These results demonstrate that the Bispectral Network is not computing the power spectrum. The ReLU Network, by contrast, computes a map that is related but not exactly equal to the power spectrum,

Performance on the noisy test dataset also begins to tease apart the two networks. The responses of the ReLU Network to noisy data reveal greater brittleness in the learned map. Figure 4.11 shows k -nearest neighbor performance of the Bispectral and ReLU Networks on the noisy test dataset, with noise levels parameterized by the standard deviation of the additive Gaussian noise distribution, ranging from $\sigma = 0.0$ to $\sigma = 0.3$. The Bispectral Network shows remarkable robustness to noise, demonstrating no k -nearest-neighbor classification error for any of the noisy datasets. The performance of the ReLU network, by contrast, degrades rapidly, with 10.59% error on the $\sigma = 0.1$ dataset, 35.82% error on $\sigma = 0.2$ and 59.53% error on the noisiest $\sigma = 0.3$ dataset.

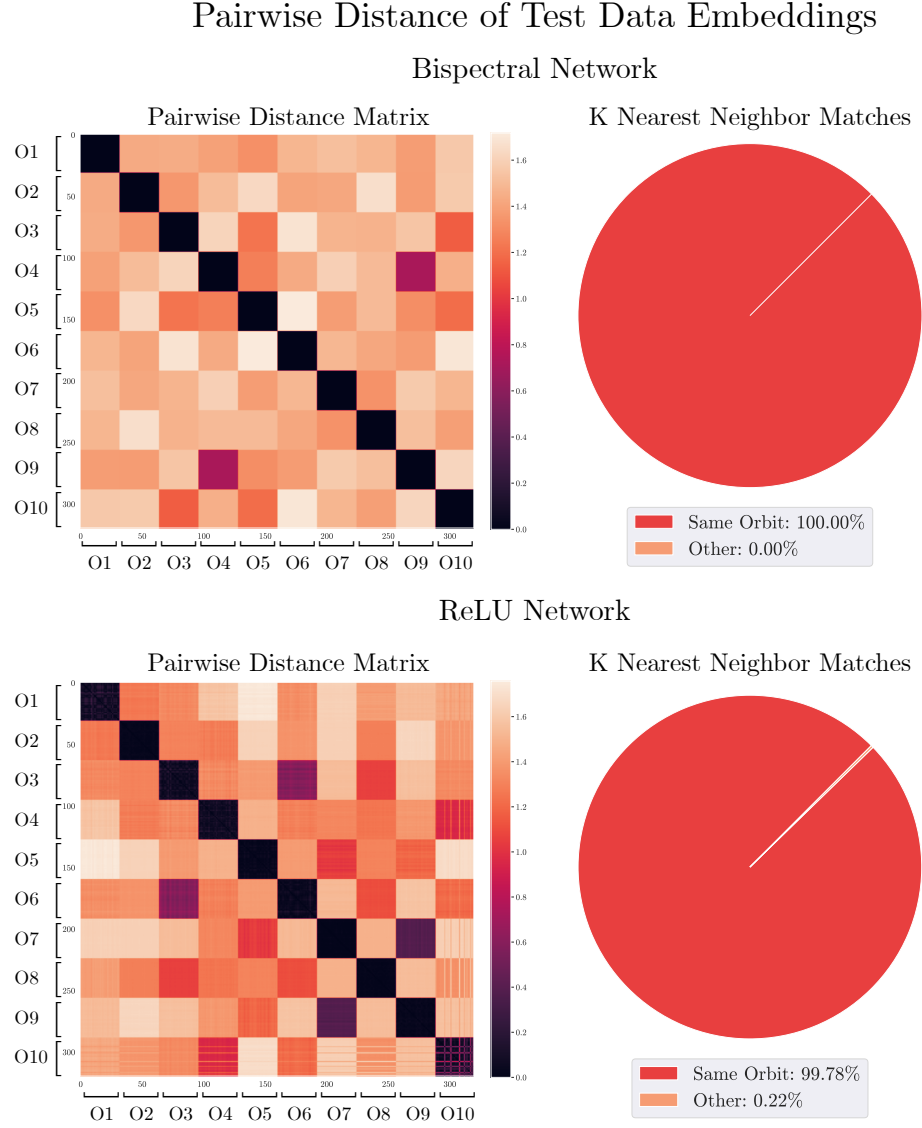
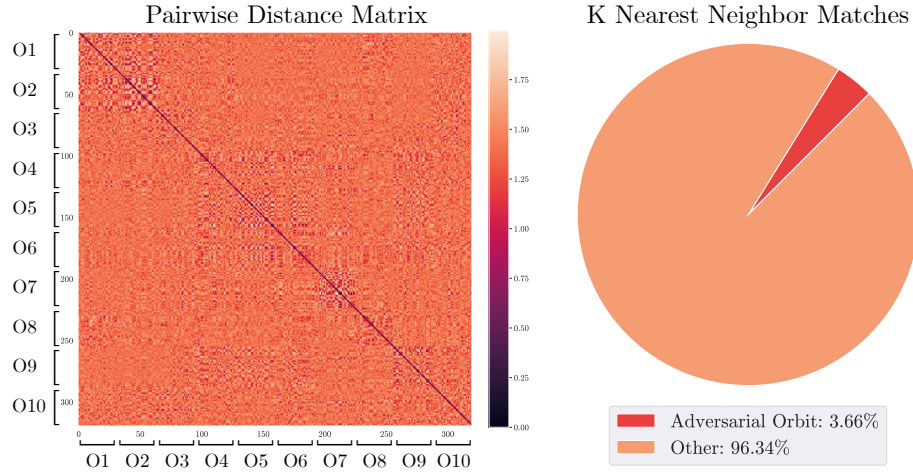


Figure 4.9: Pairwise Distance of Test Data Embeddings. (Left) Pairwise distance matrices for 10 example test orbits are shown for both the Bispectral and ReLU Networks. Each row and column in the matrix index a pair of datapoints. Datapoints are grouped by orbit, labeled by “O1”, “O2,” etc. (Right) K-nearest-neighbor analysis of the average nearest neighbor orbit membership for each data vector. Both Bispectral and ReLU Networks achieve robust generalization performance on the test set, with 100% and 99.78% correct orbit classification for each network, respectively.

Pairwise Distance of Adversarial Data Embeddings

Bispectral Network



ReLU Network

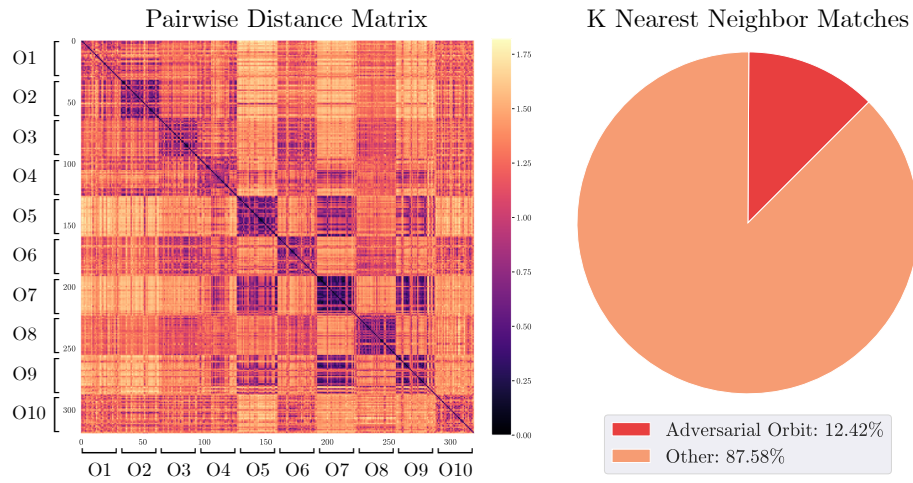


Figure 4.10: Pairwise Distance of Adversarial Data Embeddings. (Left) Pairwise distance matrices for 10 example adversarial orbits are shown for both the Bispectral and ReLU Networks. Each row and column in the matrix index a pair of datapoints. Datapoints are grouped by adversarial orbit, labeled by “O1”, “O2,” etc. (Right) K-nearest-neighbor analysis of the average nearest neighbor orbit membership for each data vector. The Bispectral Network groups adversarial orbit members together at a rate predicted by chance, 3.66%, whereas the ReLU Network groups adversarial orbit members together 12.42% of the time.

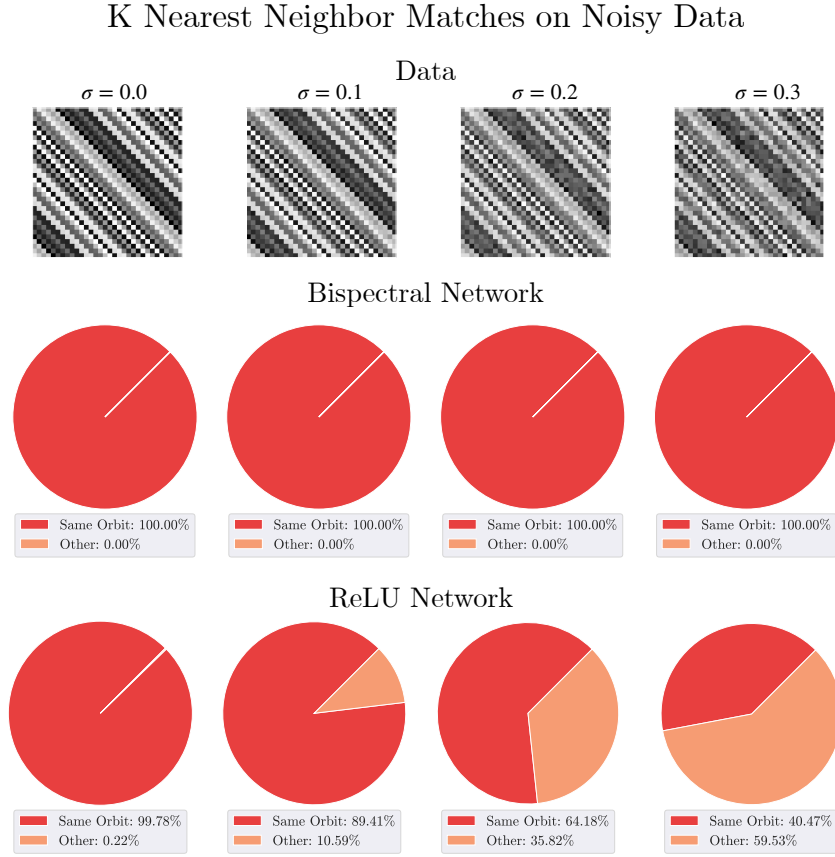


Figure 4.11: K-Nearest-Neighbor Matches on Noisy Data. The Bispectral Network (Top) and ReLU Network (Bottom) are evaluated on k-nearest-neighbor classification performance on datasets with increasing noise levels. The Bispectral Network achieves perfect (100%) classification performance for all noise levels. The ReLU Network attains considerably lower performance, with only 35.82% average correct classification on the $\sigma = 0.2$ dataset.

Summary

Both Bispectral and ReLU Networks successfully minimize the contrastive loss objective on this task, and achieve high performance on the training and validation datasets. In addition, both networks exhibit robust generalization to entirely novel test datapoints drawn from the training data distribution. Remarkably, both networks learn Fourier-based invariants, as revealed by the learned 1D Fourier basis functions in the first layer of all networks. However, further analyses reveal considerable differences in the character of the full network output. Whereas the Bispectral Network generalizes gracefully to noisy test data, the ReLU Network exhibits considerable brittleness to noise. Performance on an adversarial dataset, in which adversarial “or-

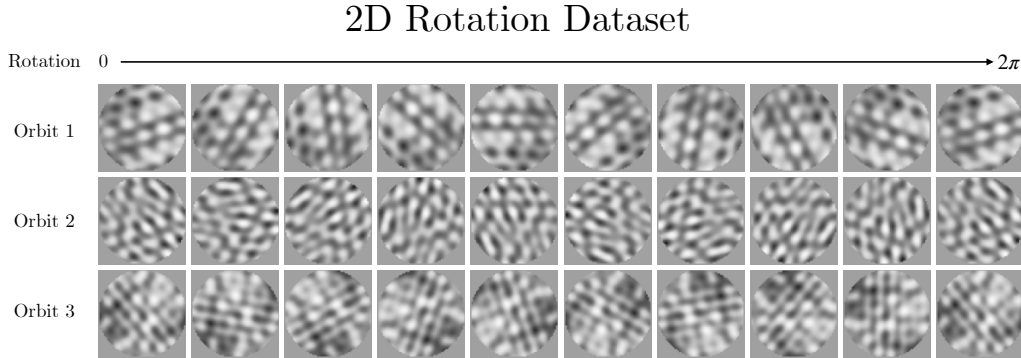


Figure 4.12: 2D Rotation Dataset. Three example patterns and their orbits are shown, with a subset of 10 rotations per orbit. Patterns are generated as a sum of plane waves with coefficients drawn from $\mathcal{U}(0, 1)$.

bits” have identical power spectra but non-equal bispectra reveals that the learned map in the ReLU Network involves, at least partially, the computation of the power spectrum.

4.4.3 2D Rotation $SO(2)$

We now consider the group of 2D rotations, defined by the set of 2×2 special orthogonal matrices $SO(2)$. Bispectral and ReLU Networks are trained on a synthetic dataset consisting of random 2D patterns and their orbits under $SO(2)$.

Methods

The training dataset consists of 100 random 2-dimensional patterns of dimension 60×60 , generated as linear combinations of a randomly selected set of 10 plane waves (2D Fourier filters) with maximum frequency of 10 and their orbits under 2D rotation. Coefficients on each filters are drawn from $\mathcal{U}(0, 1)$ before they are linearly combined. Patterns are cropped to a circle with diameter equal to the side length of the image, with values outside of the circle set to zero. The group $SO(2)$ is discretized into 36 discrete angles and the orbit of each pattern is generated as the set of all discretized rotations of the pattern. Figure 4.12 depicts three random patterns and a subset of their orbits. A random 20% of the dataset is set aside for model validation and is used to tune hyperparameters. The remaining 80% is used for training.

The test dataset consists of 100 novel random patterns and their orbits, generated with the same method as the training data. In addition, a rotated version of the Omniglot [73] dataset is used to test the capacity of the trained networks to gen-

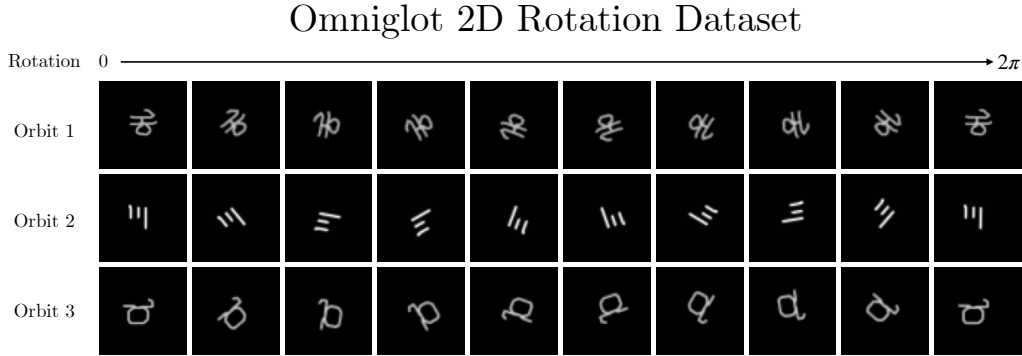


Figure 4.13: Omniglot 2D Rotation Dataset. Three example characters and their orbits are shown, with a subset of 10 rotations per orbit.

eralize from synthetic data to data with more natural statistics. For this dataset, 100 random exemplar characters are selected and their orbits are generated using the same discretization of $SO(2)$ as in the training data. Figure 4.13 shows three example Omniglot characters and a subset of their orbits.

Models were trained until convergence, which took no more than 500,000 total data examples. Each model is evaluated on its *training and validation performance*, *learned weight structure*, *invariance and equivariance of the learned maps*, and *generalization performance*.

Results

Training and Validation Performance

Training and validation performance for the best-performing ReLU and Bispectral Networks are displayed in Table 4.4. Both networks achieve low validation loss, with the ReLU Network achieving the best performance. The the best-performing ReLU Network used 1,250,996 parameters, whereas the best Bispectral Network used 691,200 parameters.

Architecture	Train Loss	Validation Loss	Parameters
ReLU	0.0007	0.0009	1,250,996
Bispectral	0.0058	0.0072	691,200

Table 4.4: Training and Validation Performance on $SO(2)$ Rotation for the best of each architecture tested. The real and imaginary components of each neuron are counted as separate parameters for the Bispectral Networks.

Filters Learned in the Bispectral Network

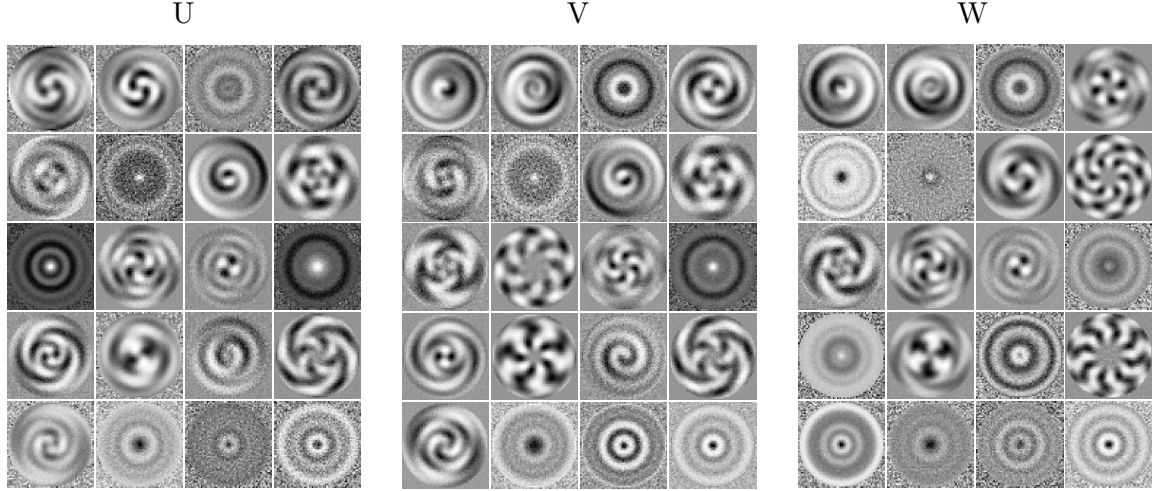


Figure 4.14: Filters Learned in the Bispectral Network Trained on the $SO(2)$ Dataset. The real components of filters in the U , V , and W weight matrices are shown here. Filters were ordered according to the average response magnitude on the test dataset and the top 20 in each matrix were selected for visualization.

Learned Weight Structure

We now examine the structure of the learned weights in both architectures. Figures 4.14 and 4.15 show the learned weights for the best-performing Bispectral and ReLU Networks, respectively. The Bispectral Network learns a diverse set of filters with concentric and spiral structure, with many filters equivalent to disk harmonics. The ReLU Network, by contrast, learns a homogeneous set of first-layer filters, exclusively consisting of concentric patterns.

Only the real component of the learned filters are shown in Figure 4.14. The real and imaginary components of the learned sinusoidal spiral filters come in quadrature pairs, in which the imaginary component is offset in phase by $\pi/2$ radians in polar coordinates. Figure ?? shows the real and imaginary components of an example filter. This property makes the output of a linear transform defined by these filters equivariant to 2D rotation in the input, resulting in a phase shift that is proportional to the frequency of each filter in polar coordinates.

Filters Learned in the ReLU Network

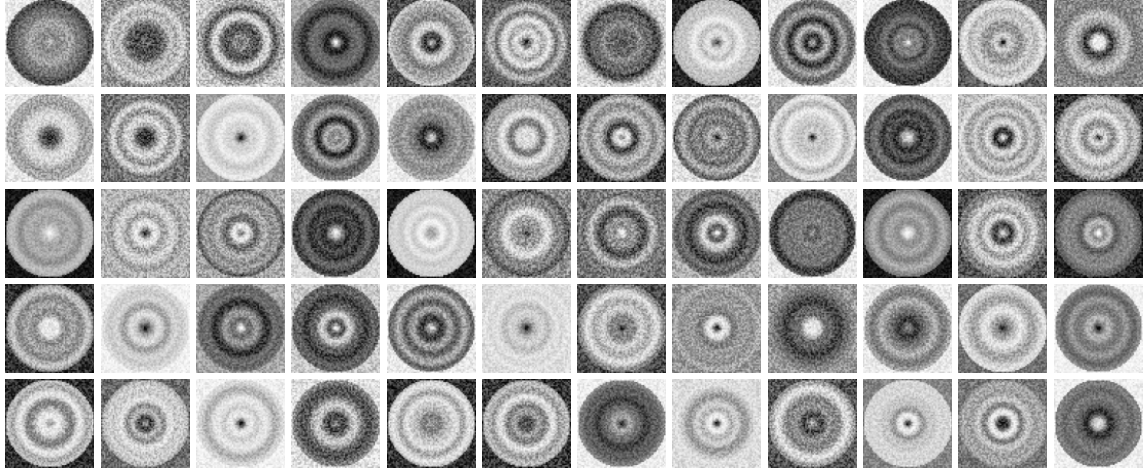


Figure 4.15: Filters Learned in the ReLU Network Trained on the $SO(2)$ Dataset. The 60 filters with highest average magnitude responses on the test dataset in the first layer are shown here.

Map Structure

The properties of these filters under rotation of the input is reflected in an examination of the output of the first linear term in the Bispectral Network. Figure 4.16 compares the outputs of the first linear term and full network output for both the Bispectral Network and the ReLU Network for three example orbits. The conventions for this plot are the same as for Figure 4.8. As the input data is continuously rotated, the outputs of each neuron in each linear term in the first layer of the Bispectral Network continuously shifts in phase—as expected of a rotation equivariant Fourier transform. The output of the full network is perfectly invariant to rotations in the input.

The output of the first linear term in the ReLU Network, by contrast, is invariant rather than equivariant to rotations in the input. This is expected, given the learned weight structure in the ReLU Network, as all of the learned weights have the invariant “target” structure rather than the equivariant “spiral” structure. All neurons in the ReLU Network are pictured in both the first linear term and full output plots in Figure 4.16, revealing that this holds for all neurons in the network.

Model Outputs on Rotating Data

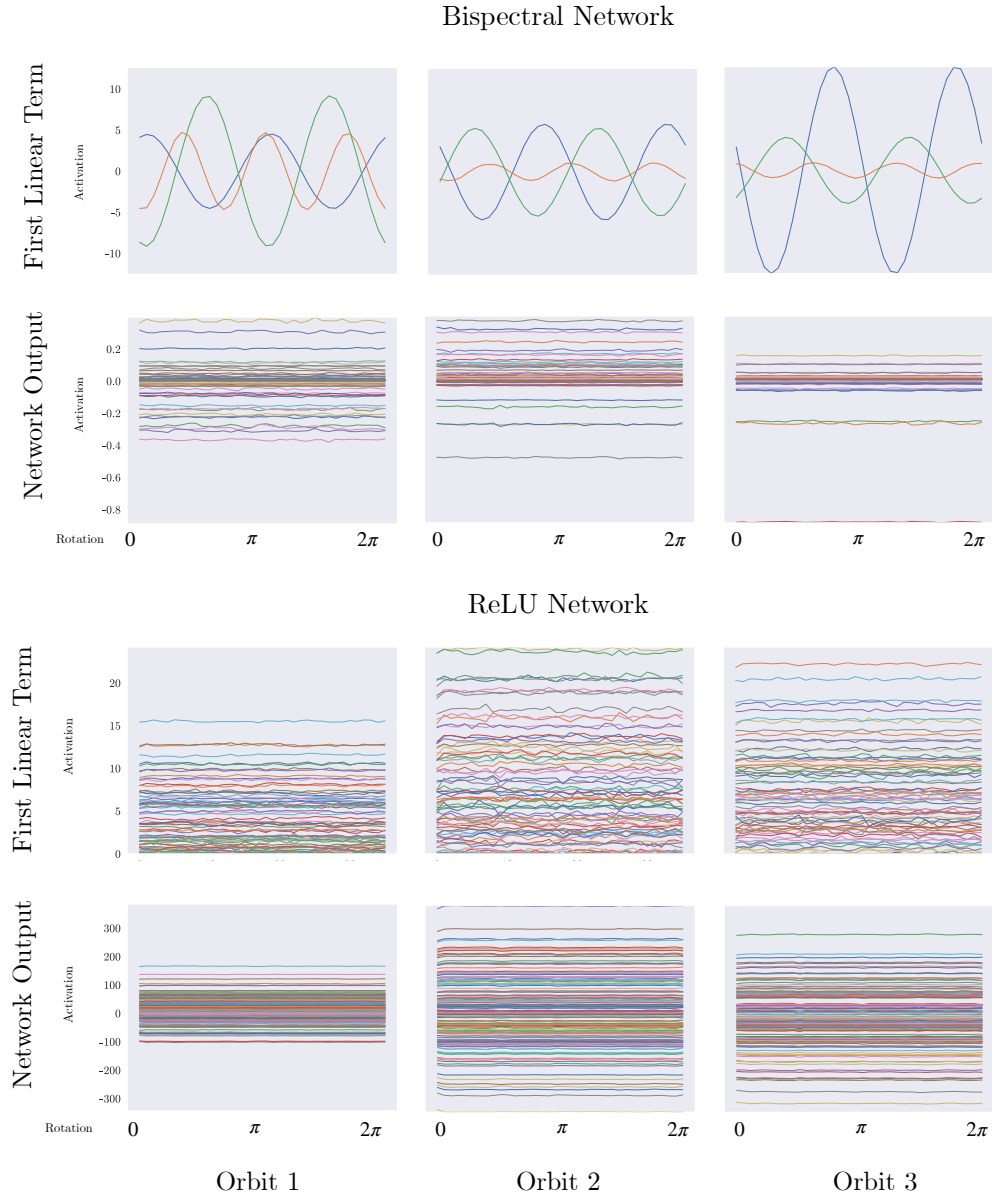


Figure 4.16: Model Outputs on Rotating Data. Outputs of the first linear term and full network output are displayed for the Bispectral Network (Top) and ReLU Network (Bottom). Data inputs are rotated (x-axis) and layer/network outputs are computed for each rotation. Each colored line represents the output of a single neuron, with the value of the activation on the y-axis.

Generalization to Novel Data

Model performance is examined on the test and rotated Omniglot datasets. Model outputs for each dataset are computed and performance is quantified by calculating the average proportion of k -nearest-neighbors of each datapoint that are within-orbit, with k equal to the size of the orbit and the Euclidean distance determining nearest neighbors. As in the previous study, we expected 100 % within-orbit nearest-neighbors for a model that has learned to selectively collapse orbits.

Figure 4.17 shows pairwise distance matrices for 10 example orbits in the test dataset. Remarkably, the Bispectral Network achieves perfect generalization performance to this entirely novel dataset, with ReLU Networks achieving with comparable near-perfect performance.

More remarkably, both networks generalize robustly to the Rotated Omniglot dataset and achieve comparable performance as on the test dataset, despite the considerable difference in signal statistics. Figure 4.18 shows distance matrices for 10 example orbits and k -nearest-neighbor results for the full 100 orbit Rotated Omniglot dataset. The Bispectrum Network achieves an average of 99.81% within-orbit neighbors per data sample. The ReLU Network performs comparably well albeit slightly worse, with an average of 98.51% within-orbit neighbors.

However, the performance of the two networks begins to tease apart when tested on noisy data. Figure 4.19 shows k -nearest-neighbor analyses for the noisy test datasets, for three levels of increasing noise. The Bispectral Network achieves comparable performance on all 4 datasets, with the noisiest dataset ($\sigma = 0.3$) resulting in k -nearest-neighbor accuracy of 98.39%. The performance of the ReLU Network, by contrast, breaks down at a much lower noise level, resulting in accuracy of 80.93% for $\sigma = 0.2$ and 63.74% for $\sigma = 0.3$. This result mirrors the pattern observed in the cyclic translation study.

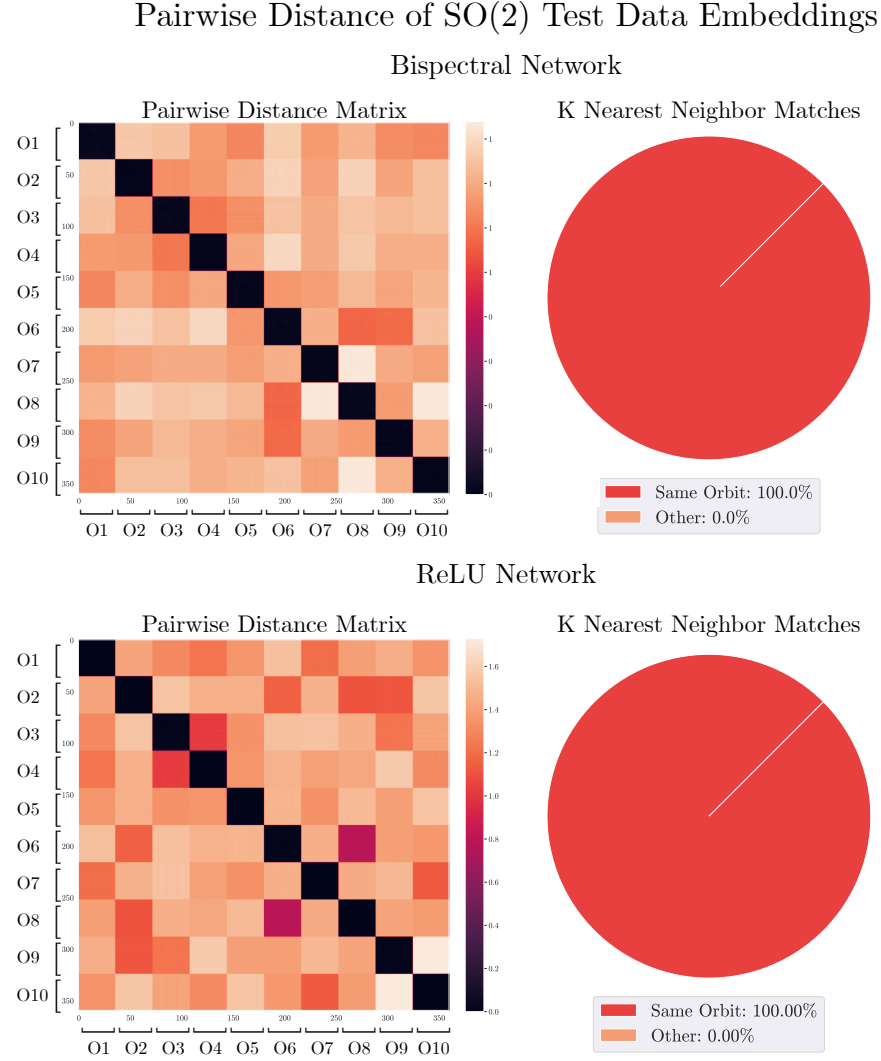


Figure 4.17: Pairwise Distance of $SO(2)$ Test Data Embeddings. (Left) Pairwise distance matrices for 10 example test orbits are shown for both the Bispectral and ReLU Networks. Each row and column in the matrix index a pair of datapoints. Datapoints are grouped by orbit, labeled by “O1”, “O2,” etc. (Right) K-nearest-neighbor analysis of the average nearest neighbor orbit membership for each data vector. Both Bispectral and ReLU Networks achieve robust, perfect generalization performance on the test set, with 100% correct orbit classification for both.

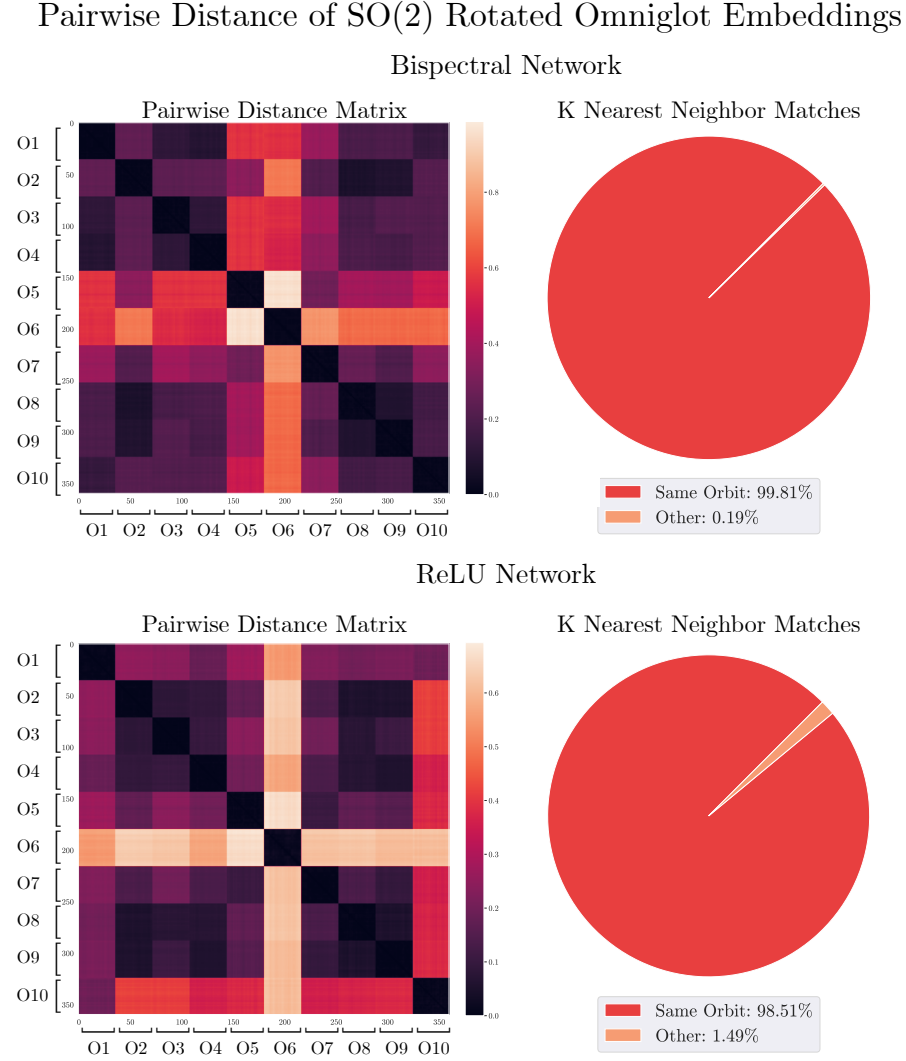


Figure 4.18: Pairwise Distance of $SO(2)$ Rotated Omniglot Embeddings. (Left) Pairwise distance matrices for 10 example test orbits are shown for both the Bispectral and ReLU Networks. Each row and column in the matrix index a pair of datapoints. Datapoints are grouped by orbit, labeled by “O1”, “O2,” etc. (Right) K-nearest-neighbor analysis of the average nearest neighbor orbit membership for each data vector. Both Bispectral and ReLU Networks achieve robust, perfect generalization performance on the test set, with 100% correct orbit classification for both.

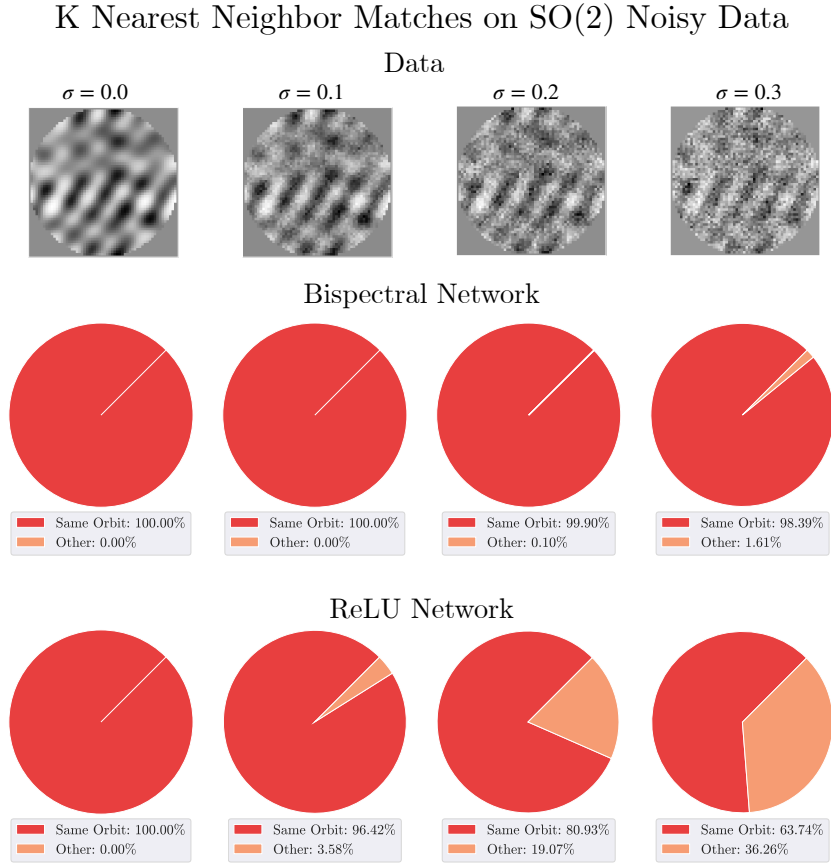


Figure 4.19: K-Nearest-Neighbor Matches on SO(2) Noisy Data. The Bispectral Network (Top) and ReLU Network (Bottom) are evaluated on k-nearest-neighbor classification performance on datasets with increasing noise levels. The Bispectral Network achieves perfect or near-perfect classification performance for all noise levels, with the highest noise level ($\sigma = 0.3$) resulting in 98.39% average correct classification. The ReLU Network attains considerably lower performance, with 80.93% average correct classification on the $\sigma = 0.2$ dataset and 63.74% correct on the $\sigma = 0.3$ dataset.

Summary

Both Bispectral and ReLU Networks achieve high performance on the training, validation, test, and Rotated Omniglot datasets. However, the structure of the weights learned in these networks reveal distinct strategies. The Bispectral Network learns rotationally equivariant filters that are related to the harmonics of the disk. The ReLU Network, by contrast, learns only rotationally invariant filters, thus losing the transformation information after the first stage of processing. The consequences of these two strategies are quantitatively teased apart by examining the networks' per-

formance on noisy test data, which reveals considerable brittleness in the learned ReLU Network solution.

4.4.4 3D Rotation $SO(3)$

We now tackle the more complex problem of learning invariance to a non-commutative group. Here, we consider the group $SO(3)$ defined by the set of special orthogonal matrices of dimension 3, which define the actions of rotation on a sphere. For this study, input signals are defined as functions on the 2-Sphere, represented in spherical coordinates. Concretely, a signal is a map $x : S^2 \rightarrow \mathbb{R}$, sampled at a grid of dimension $(L+1) \times (2L+1)$, defined by the $(L+1)$ zeros for the Legendre Polynomial of degree $(L+1)$, using the approach associated with the exact Gauss-Legendre quadrature [103]. The orbit of the signal is approximated by a discrete sampling of $SO(3)$. The implicit coordinates of the learned weights are spherical, but this is imposed by no structure other than the relationships between transformed datapoints.

Learning Non-Commutative Groups

The irreducible representations of all commutative groups are scalar-valued. This is not generally true of non-commutative groups, which may have matrix irreps of arbitrary dimension. Consequently, the form of the generalized bispectrum for non-commutative groups must be altered, as the outputs of projections onto the irreps of the group must be combined through the tensor product and then block diagonalized using the Clebsch-Gordan coefficients, as described in Equation 3.9 and reproduced here for clarity:

$$\beta_{\rho_1, \rho_2} = [\hat{f}_{\rho_1} \otimes \hat{f}_{\rho_2}] C_{\rho_1, \rho_2} \left[\bigoplus_{\rho \in \rho_1 \otimes \rho_2} \hat{f}_{\rho}^{\dagger} \right] C_{\rho_1, \rho_2}^{\dagger},$$

with C a Clebsch-Gordan matrix, $\otimes$ the tensor product, and $\oplus$ a direct sum over irreps. Consequently, the ansatz used to define the Bispectral Network must change to equip the network with the computational capacity to perform this more complex operation. Noting that this calculation requires an additional second order operation (the tensor product), we define the Non-Commutative Bispectral Network in two layers as:

$$T(x) = Ux \odot Vx \tag{4.4}$$

$$B_{NC}(x) = U'T(x) \odot V'T(x) \odot W'T(x) \tag{4.5}$$

We expect the first layer weights in U and V to learn the irreducible representations of the group—which would be the spherical harmonics for this study. The matrices in the second layer, U' , V' , and W' should learn the changes of basis on the tensor product required to achieve bispectral phase cancellation.

Methods

The training dataset consists of 100 random signals defined on the sphere, generated as linear combinations of spherical harmonics with coefficients drawn from $\mathcal{U}(0, 1)$. The sphere is discretized into a 32×64 Gauss-Legendre Quadrature grid. We generate a subset of each datapoint's orbit under the group $SO(3)$, selecting 200 random discrete rotations for each datapoint drawn uniformly from the set of all rotations under the discretization scheme used. A random 20% of the dataset is set aside for model validation and is used to tune hyperparameters. The remaining 80% is used for training. The test dataset consists of 100 novel datapoints and their orbits generated in the same manner. Figure 4.20 shows three example datapoints and a subset of the rotations in their orbits. Here, we train just the Non-Commutative Bispectral Network on the task.

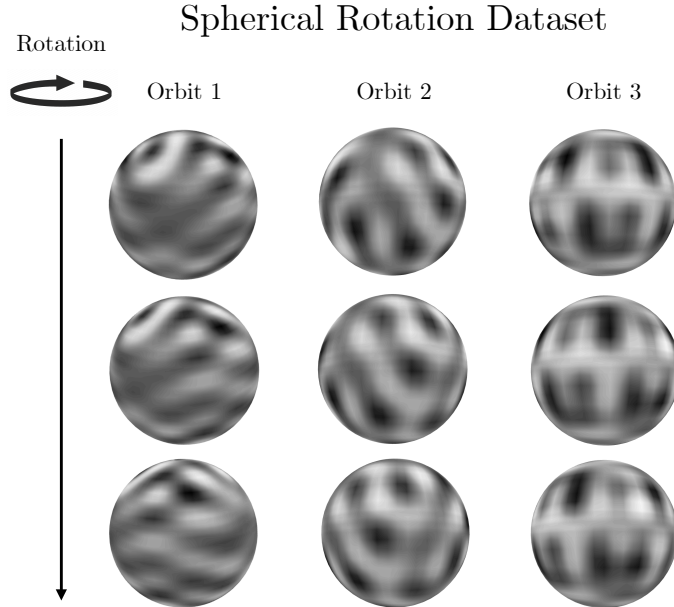


Figure 4.20: Spherical Rotation Dataset. Three example signals and their orbits are shown, with a subset of 3 rotations per orbit. Patterns are generated as a linear combination of spherical harmonics with coefficients drawn from $\mathcal{U}(0, 1)$.

Results

Table 4.5 shows Training and Validation performance for the Non-Commutative Bispectral Networks. The best-performing Bispectral Network has 512 complex-valued neurons in each weight matrix in both layers, resulting in a total of 5,767,168 parameters and achieves a loss of 0.024.

Architecture	Train Loss	Validation Loss	Parameters
Non-Commutative Bispectral	0.0187	0.0240	5,767,168

Table 4.5: Training and Validation Performance on $SO(3)$ Rotation for the best of each architecture tested. The real and imaginary components of each neuron are counted as separate parameters for the Bispectral Networks.

Remarkably, the Bispectral Network learns weights that approximate the spherical harmonics. Figure 4.21 shows six example weights learned in the Bispectral Network. Each weight is visualized as a function on the surface of the sphere, with the distance from the center of the sphere indicating the magnitude of the function and the color indicating the sign.

Weights Learned in the Bispectral Network on $SO(3)$

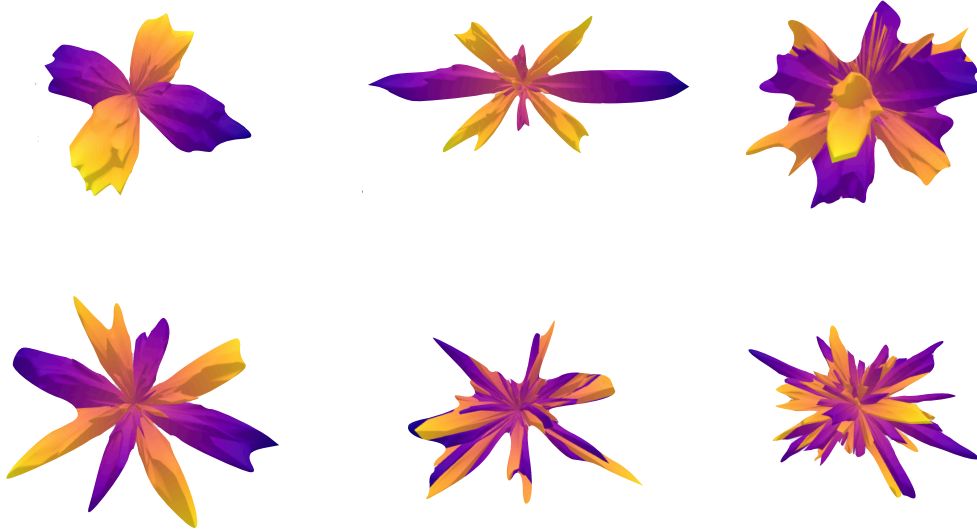


Figure 4.21: Weights Learned in the Bispectral Network on $SO(3)$. Six example weights are visualized as functions on the surface of the sphere, with the distance from the center of the sphere indicating the magnitude of the function and the color indicating the sign.

Moreover, network generalizes robustly to novel data drawn from the same distribution. On a test set of 100 novel patterns and their orbits, the Bispectral Network successfully collapses 100% of the datapoints into orbit classes, as revealed by the k-nearest-neighbor analysis. Figure 4.22 shows a Euclidean distance matrix for the first 10 of those novel classes.

Pairwise Distance of $SO(3)$ Test Data Embeddings

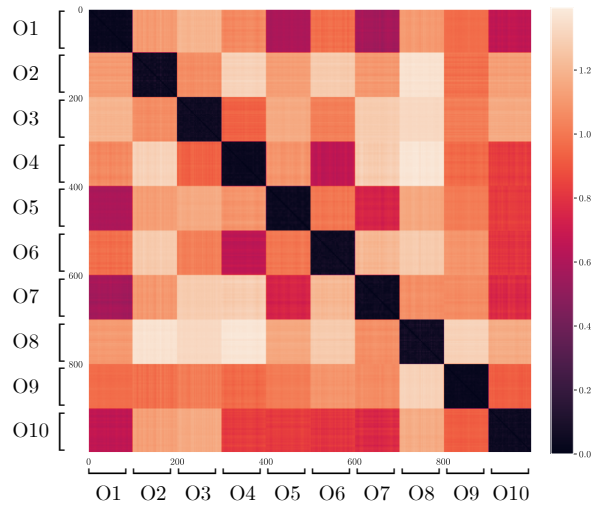


Figure 4.22: Pairwise Distance of on $SO(3)$ Test Data Embeddings. Each row and column in the matrix index a pair of datapoints and the value indicates their Euclidean distance. Datapoints are grouped by orbit, labeled by “O1”, “O2,” etc.

Additionally, the first linear term of the network has the equivariant structure observed in other Bispectral Networks, as depicted in Figure 4.23. To generate this plot, test data were generated using a linear grid of rotation angles on a single axis. These findings suggest the potential of the Bispectral Network for learning arbitrary non-commutative groups.

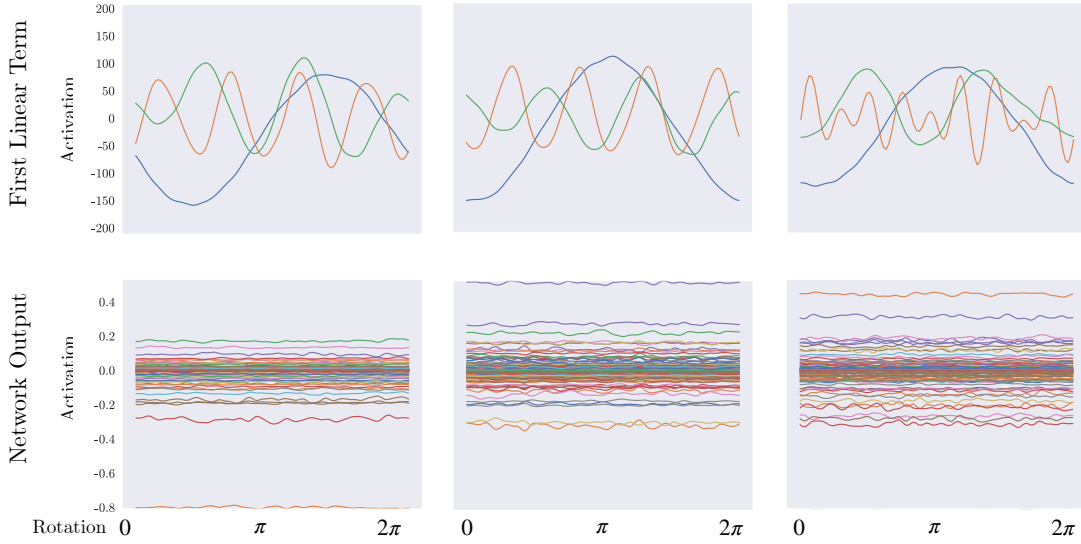
Bispectral Network Outputs on $SO(3)$ Rotating Data

Figure 4.23: Bispectral Network Output on $SO(3)$ Rotating Data. Outputs of the first linear term and full network output are displayed. Data inputs are rotated around a single axis on the sphere and layer/network outputs are computed for each rotation. Each colored line represents the output of a single neuron, with the value of the activation on the y-axis.

4.5 Summary and Conclusions

Across groups—both commutative and non-commutative—we find that the Bispectral Network learns robust, group-invariant embeddings. Moreover, it learns approximations to canonical, complete solutions to the problem, by learning to couple group representations to cancel the equivariant phase shifts due to transformations in the input. The first linear form learns to approximate a generalized Fourier transform and the output appears to approximate a generalized Bispectrum. Consequently, the resulting trained models possess both the *invariant* properties of the bispectrum and the *equivariant* properties of the Fourier transform. In comparison to generic multi-layer ReLU architectures, the Bispectral Network demonstrates considerably stronger selectivity and robustness. Here, we have primarily analyzed the model’s performance on synthetic data. We plan to extend this approach to more naturalistic datasets in future work.

Chapter 5

Group Invariant Associative Memory

In this chapter, I introduce a complex-valued phasor associative memory network based on the bispectrum that can store and denoise patterns invariant to the actions of compact groups. The Bispectral Hopfield Network generalizes the learning rule of complex phasor networks [92] to mirror the third-order group-invariant bispectrum, resulting in an associative memory network that only needs to store a single pattern to denoise its entire orbit under the action of a group. Due to the general nature of the bispectrum, this model can be constructed for arbitrary commutative and non-commutative compact groups. Here, I demonstrate it for the group of cyclic 2D translations $\mathbb{S}^1 \times \mathbb{S}^1$.

5.1 The Problem: Group Invariant Memory and Denoising

Hopfield networks [51] have long attracted interest as a model of associative memory. In a Hopfield network, memories can be understood as basins of attraction in an energy landscape. Corrupted patterns have higher energy than the original stored patterns and can be denoised by descending the energy landscape. This perspective on associative memory has inspired decades of research, and has sustained its relevance in contemporary deep learning [106].

Hopfield network denoising can be viewed as a process of canonicalization—removing corruptions to achieve a noise-invariant signal representation. While invariant to perturbations of a statistical character, Hopfield networks are *not* invariant to perturbations of a more geometric variety. Natural transformations on the signal, such as translation or rotation, will not in general map to the same basin of attraction as the original signal. Theorists have long noted this brittleness; however, to date, a

solution has not been proposed. Here, I demonstrate that a simple modification to the a Noest phasor associative memory network [92] results in a network that only needs to store a single pattern to denoise its entire orbit under the action of a group.

Hopfield Networks

The classical Hopfield network model [51] is a single-layer recurrent artificial neural network with a set of neurons $\{z_i, \dots, z_n\}$ taking binary states in $\{-1, +1\}$. Connections between weights in the layer are specified with a matrix of synaptic strengths $W_{i,j} \in \mathbb{R}^{n \times n}$. A set of c binary patterns $\{x^1, \dots, x^c\}$, with $x^c \in \{-1, 1\}^n$ may be stored in the synapses between neurons, using the *outer product learning rule*:

$$W_{i,j} = \sum_c x_i^c x_j^c, \quad (5.1)$$

that is, as the sum of each pattern's outer product with itself. These patterns define basins of attraction in the energy landscape of the system. The *energy* of a given network is calculated as:

$$E(z) = -\frac{1}{2} \sum_{i,j} W_{i,j} z_i z_j. \quad (5.2)$$

This specifies a Lyapunov function for the system, which can be descended with the following dynamics:

$$z_i = \begin{cases} +1 & \sum_j W_{i,j} z_j > 0 \\ -1 & \text{otherwise} \end{cases}$$

The concept was extended to complex-valued states in Noest's phasor networks [92].

Phasor Networks

In a phasor network, neurons take states $z \in \mathbb{C}$. Generalizing the generic outer product learning rule, the weight matrix of a phasor network is defined:

$$W_{i,j} = \sum_c x_i^c x_j^{c\dagger} \quad (5.3)$$

The energy of a network state z is:

$$E(z) = -\frac{1}{2} \sum_{i,j} W_{i,j} z_i z_j^\dagger \quad (5.4)$$

The network state is updated as the angle of the dot product of the current state with the weight matrix:

$$u_i = \sum_j W_{i,j} z_j \quad (5.5)$$

$$z_i = \frac{u_i}{|u_i|} \quad (5.6)$$

The model has been extended to the sparse regime in [34]. In the existing literature, phasor associative memory models have been used primarily to store random patterns of phasors. The sparse phasor network of [34] was demonstrated on the task of denoising natural images; however, the images were first mapped to random phasor patterns through a codebook. We propose, alternatively, to store the *phase spectra of signals*, with each neuron encoding the phase of a complex Fourier coefficient in frequency space. This perspective allows us to define a novel phasor associative memory network that uses a simple modification of the outer product learning rule, which we call the *bispectral learning rule*.

5.2 Model: Bispectral Hopfield Networks

We generalize the learning rule and energy function of Phasor Networks to define the Bispectral Hopfield Network. To store vector phasor patterns $\mathbf{x}^1, \dots, \mathbf{x}^c \in \mathbb{C}^d$, the weight matrix W is computed as a superposition of the bispectra of each pattern:

$$W_{i,j} = \sum_c x_i^c x_j^c x_{i+j}^{c\dagger}, \quad (5.7)$$

with the index sum $i + j$ computed modulo the vector dimension d , and i, j indexing ordered Fourier frequencies. Effectively, this stores the relative phase structure of the signal, while eliminating absolute phase. Notably, while the network structure makes use of third-order interactions, it uses the same number of weights as the generic phasor network. This is due to the constraint on the third term, in which the index ranges only over the values $k = i + j$. Note that the weight matrix is symmetric but not Hermitian. The energy of a network state z is defined as:

$$E(z) = -\left| \sum_{i,j} W_{i,j} z_i^\dagger z_j^\dagger z_{i+j} \right|, \quad (5.8)$$

which is the negative magnitude of the inner product between the weight matrix and the bispectrum of the state. The energy can be descended with the network dynamics:

$$u_k = \sum_j W_{k,j} z_j z_{k+j}^\dagger + \sum_i W_{i,k} z_i z_{i+k}^\dagger + \sum_{i+j=k} W_{i,j} z_i z_k, \quad (5.9)$$

$$z_k = \frac{u_k}{|u_k|}, \quad (5.10)$$

preserving only the phase of the state at each update. Due to the translation invariance of the bispectrum, storing a single exemplar of a pattern enables the network to denoise its entire orbit under cyclic translation.

The above equations define a 1-dimensional translation invariant Bispectral Hopfield Network. This can be extended to 2-dimensional cyclic translation $\mathbb{S}^1 \times \mathbb{S}^1$ as follows. Given a set of images $X^i, \dots, X^c \in \mathbb{R}^{d \times d}$, the storage rule is defined:

$$W_{i_1, i_2, j_1, j_2} = \sum_c x_{i_1, i_2}^c x_{j_1, j_2}^c x_{i_1+j_1, i_2+j_2}^{c\dagger}, \quad (5.11)$$

which is the superposition of the 2-dimensional bispectra of the signals. The energy function and update equations are modified accordingly. The synaptic weights W_{i_1, i_2, j_1, j_2} are now specified as a 4-tensor. However, this does not increase the dimensionality of the network; a signal of dimension $N = n \times n$ is stored in $N = n \times n$ neurons.

Here, we examine the properties of the Bispectral Hopfield Network in two contexts. In Experiment 1, we test the memory capacity of the network for storing random phasor images and their translations. In Experiment 2, we test the capacity of the network to denoise natural images under a variety of different perturbations.

5.3 Experiment 1: Memory Capacity for Random Phasor Patterns and Their Translations

We first examine the storage capacity of the Bispectral Hopfield Network for signals containing no correlation structure—i.e. random phasor patterns. We use the 2-dimensional form of the Bispectral Hopfield Network, for denoising 2-dimensional images. A set of canonical patterns is stored in the network weights. We then test the model’s capacity to denoise noisy versions of both the stored patterns and the noisy orbits of those patterns under the action of cyclic 2D translation.

5.3.1 Methods

A set of random phasor images $X^1, \dots, X^c \in \mathbb{C}^{d \times d}$ with $|X_i^c| = 1$ are generated as patterns of phasor angles drawn from $\mathcal{U}(-\pi, \pi)$. Here, we generate 2-dimensional phasor images of shape 20×20 and use a Bispectral Network with 400 (20×20) neurons. The phasor images can be thought to specify the phase spectra of an image in pixel space, which can be produced from the phasor patterns through the inverse Fourier transform. The orbit of each pattern on cyclic 2D translation is generated in single pixel increments in pixel space and converted back to phase space for a total of 400 translations per phasor image and a total dataset size of $400c$. Figure 5.1 shows a set of example patterns and their orbits in both pixel space (top) and phase space (bottom).

We generate datasets with 10 - 150 classes, in increments of 10, and examine how the network’s denoising abilities degrade as the number of classes grows. For this experiment, only a single exemplar from each orbit is stored in the network weights. We then examine the network’s ability to denoise noisy versions of the stored patterns and their orbit. Noise is added in phase space and denoised patterns are evaluated in pixel space, using the cosine distance from the original image. Here, we use additive von Mises noise to the phase angles, with $\kappa = 1$, which results in noisy patterns with an average cosine distance of 0.45 from the original patterns in pixel space.

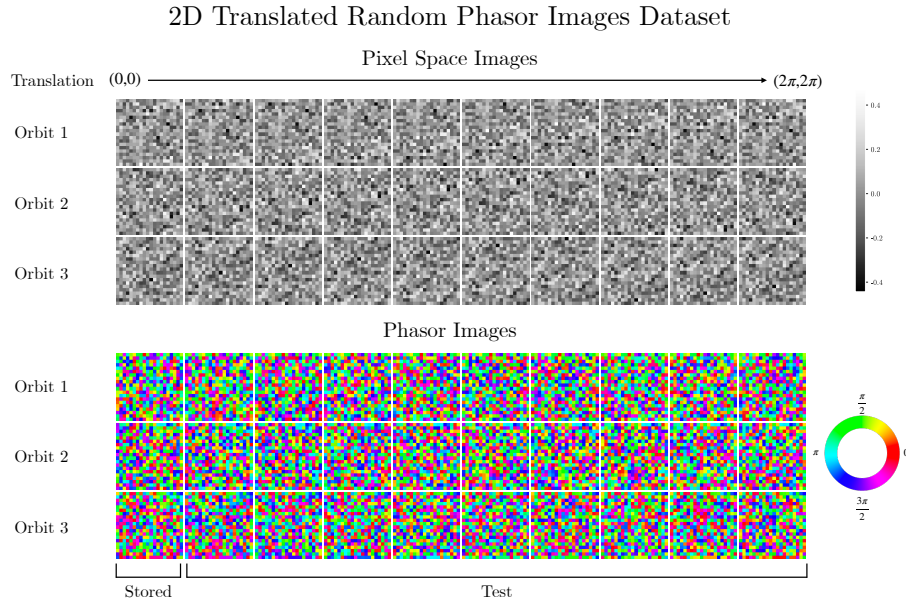


Figure 5.1: 2D Translated Random Phasor Images Dataset. Three example orbits are displayed in pixel space (top) and phase space (bottom). Only a single exemplar from each orbit is stored.

5.3.2 Results

Figure 5.2 (top) shows the cosine similarity of the denoised patterns with the originals, as the number of classes is ranged from 10 to 150. The plot includes both stored patterns and their orbits and demonstrates robust denoising for all patterns at small dataset sizes, with performance beginning to drop off around 110 patterns and degrading entirely for 130 patterns. This suggests a memory capacity of approximately $0.3N$ *classes*, with $N = n \times n$ the number of neurons in the network. Notably, the capacity is defined over *classes* rather than *exemplars*. Given the structure of the model, it denoises the entire orbit of the patterns “for free,” effectively increasing the memory capacity by the size of the group (here by a multiplicative factor of N). This results in a memory capacity of $0.3N^2$ exemplars.

Figure 5.2 (bottom) classifies patterns as either successfully denoised if the cosine similarity to the true pattern is ≥ 0.9 or unsuccessfully denoised if it is not. We then calculate the percentage of successfully denoised patterns across the dataset, averaged over the stored patterns and their unseen orbits separately. The model’s identical performance across the two sets demonstrates that it generalizes robustly to all translations of the stored patterns.

Random Phasor Orbit Denoising Performance



Figure 5.2: Random Phasor Orbit Denoising Performance. (Top) Cosine similarity of denoised patterns for the orbit of transformations of the stored patterns. (Bottom) Percent success (cosine similarity ≥ 0.9) at denoising stored patterns (blue) and their transformation orbits. Bispectral Hopfield Networks successfully generalize to all transformed versions of the stored patterns, with denoising performance on the test orbit mirroring performance on the stored patterns.

5.3.3 Discussion

This experiment establishes the capacity of Bispectral Hopfield Networks to generalize to the unseen orbits of stored random patterns with uncorrelated structure and suggests a memory capacity of $0.3D^2$, which substantially extends the classical Hopfield network’s capacity of $0.138D$. We next test the network’s performance at the task of storing the phase structure of natural images.

5.4 Experiment 2: Translation Invariant Denoising of Phase Structure in Natural Images

Although phasor networks have been around for decades, to our knowledge, they have not yet been used to store the phase structure of natural signals, instead primarily demonstrated on random phasor patterns. Here, we store the phase spectra of natural images as basins of attraction in the Bispectral Hopfield Network and examine the denoising properties of the network on the translation orbits of these signals under increasing levels of noise.

5.4.1 Methods

We demonstrate the approach on image patches drawn from a dataset of high-quality images of natural textures and scenes. Patches of shape 60×60 are randomly sampled from these images, and converted to the Fourier domain. The magnitude in Fourier space is then divided out, leaving a pattern of unit-length phasor coefficients of Fourier frequencies. This can be thought of as a form of whitening. Figure 5.3 shows sample patches in phase space and pixel space.

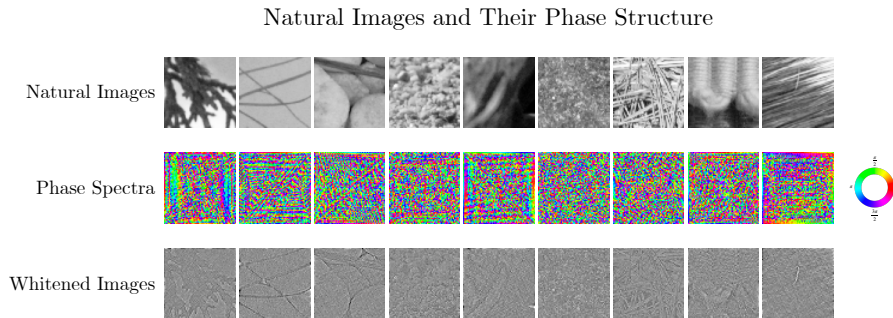


Figure 5.3: Natural Images and Their Phase Structure. Natural images (top row) are shown with their phase spectra (middle row) and the “whitened” images, generated by computing the inverse Fourier transform of the phase spectra.

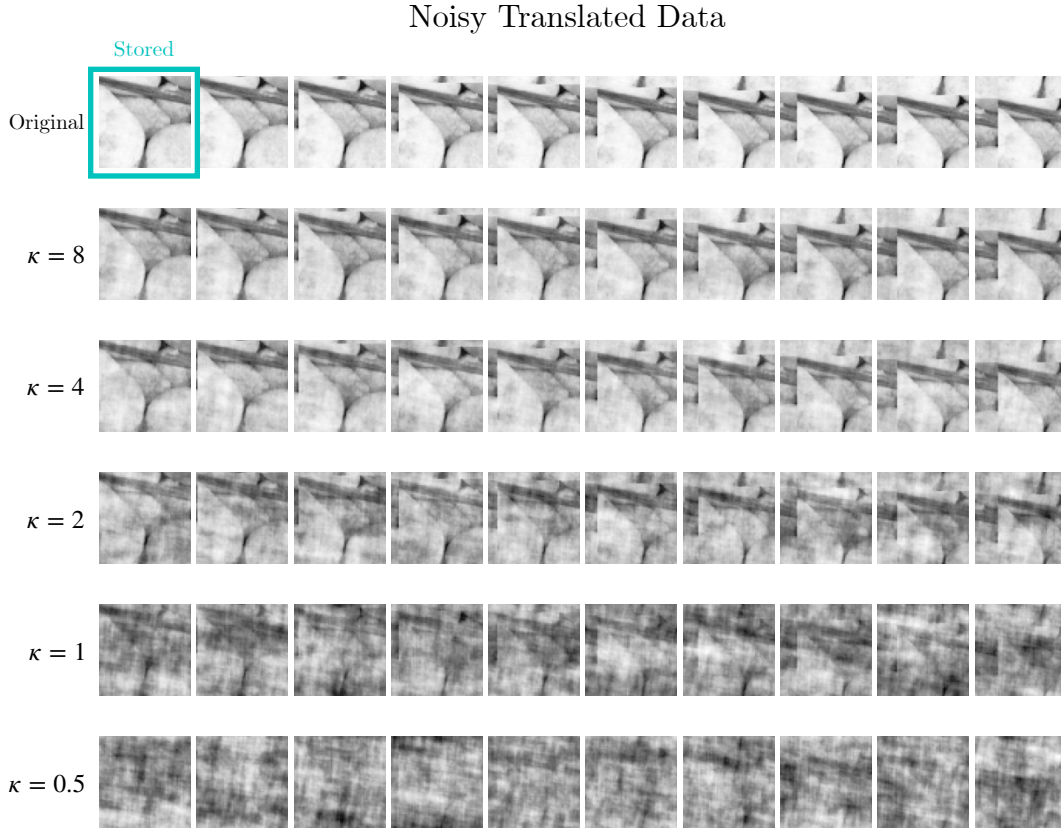


Figure 5.4: Noisy Translated Data. The translation orbit of an example natural image patch is shown under levels of increasing von Mises noise added in phase space.

For each image and its orbit, we generate a set of increasingly noisy versions, by adding von Mises noise in phase space, with κ taking values in $[8, 4, 2, 1, 0.5]$. Figure 5.4 shows example data. Only a single element of the orbit is stored. Network dynamics are run on each level of noise, for the full orbits.

5.4.2 Results

Figure 5.5 shows the result of running the Bispectral Hopfield Network on the images shown in Figure 5.4. The network perfectly denoises all translated versions of the image patches, at all levels of noise. Images are displayed with the magnitude added before inverting, for visualization purposes.

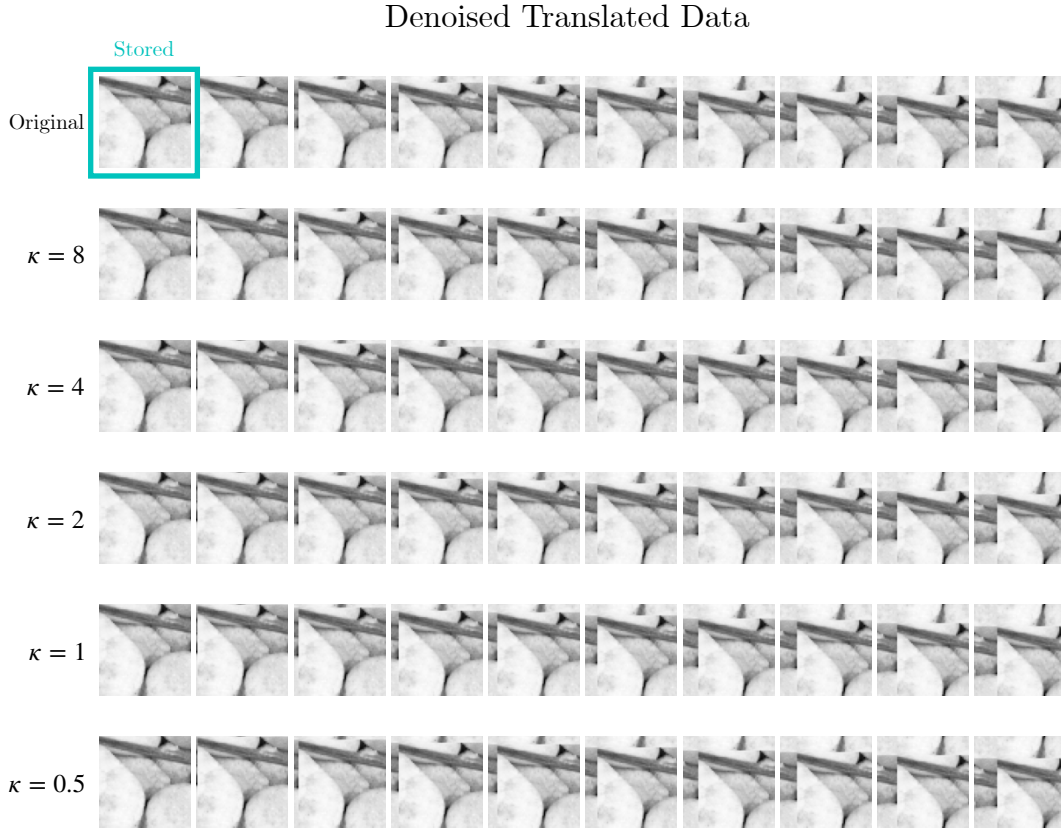


Figure 5.5: Denoise Translated Data. The denoised outputs of the Bispectral Hopfield Network. Despite only storing a single exemplar (top left), the network successfully denoises all translations of the pattern, at all noise levels.

5.4.3 Discussion

This example demonstrates the power of this approach for modeling group-invariant natural signal statistics in a dynamical system.

5.5 Summary and Conclusions

In this chapter, I introduced a novel phasor associative memory network, the Bispectral Hopfield Network that modifies the learning rule of generic phasor networks [92] to store superpositions of the bispectra of signals. This small modification to the learning rule incorporates the third order statistics of the signal while requiring no increase in the dimensionality of the network. Moreover, due to the translation-invariant na-

ture of the bispectrum, this provides a means to generalize to all translated versions of a pattern after storing only a single exemplar. This is a first step towards a more general approach using the dynamics of systems to encode the symmetries of natural signals.

Chapter 6

Discussion

Across sensory and motor regions in the brain, a striking property of neural circuits is that they tend to mirror the transformation structure of the systems they represent—either in the topological layout of their connections, or in the implicit manifold generated by their activity. This phenomenon can be observed in the circuit of neurons representing head direction in the fly [41], in the activities of grid cells [87], in the manifold structure underlying population activity in motor cortex [39], in the tonotopic organization of the cochlea [79], and in the retinotopic organization of the visual cortex [26]. This thesis advocates for a group theoretic approach to understanding and modeling these phenomena—as part of a more general computational principle of *group equivariant coding*, which enables the brain to preserve the topology of transformation structure in signals throughout stages of information processing and allows robust invariants to be computed through simple computations in the neural representation space.

The theoretical framework of efficient coding [5]—and its mathematical foundations in information and probability theory—has provided decades of immense value to the theory of neural computation. This compression-centric view has also struggled to connect coding principles with the higher-level *goals* of the organism [4]. Here, I make the case for placing the mid-level goal of *modeling the transformation structure of the natural world* at the center of neural coding theory. The algebraic and geometric properties of neural codes can have perceptual and behavioral implications for an organism. By formalizing these properties, we may begin to get a handle on the rich and often confusing observations arising from low-level neural circuitry.

This thesis offers first steps towards formalizing these properties with group theory, representation theory, and harmonic analysis, and demonstrates their utility for designing more robust, selective, and invariant signal processing algorithms. My ul-

timate goal is to bridge this work with empirical neurophysiology, towards a deeper understanding of sensory coding and the goals of perception.

Bibliography

- [1] Edward H Adelson and James R Bergen. “Spatiotemporal energy models for the perception of motion”. In: *Josa a* 2.2 (1985), pp. 284–299.
- [2] Charles H Anderson and David C Van Essen. “Shifter circuits: a computational strategy for dynamic aspects of visual processing”. In: *Proceedings of the National Academy of Sciences* 84.17 (1987), pp. 6297–6301.
- [3] David W Arathorn. *Map-seeking circuits in visual cognition: A computational mechanism for biological and machine vision*. Stanford University Press, 2002.
- [4] Horace Barlow. “Redundancy reduction revisited”. In: *Network: computation in neural systems* 12.3 (2001), pp. 241–253.
- [5] Horace B Barlow et al. “Possible principles underlying the transformation of sensory messages”. In: *Sensory communication* 1 (1961), pp. 217–234.
- [6] Andrew R Barron. “Approximation and estimation bounds for artificial neural networks”. In: *Machine learning* 14.1 (1994), pp. 115–133.
- [7] Tristan Bepler et al. “Explicitly disentangling image content from translation and rotation with spatial-VAE”. In: (Sept. 2019).
- [8] Battista Biggio et al. “Evasion attacks against machine learning at test time”. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402.
- [9] D Brillinger. “Some history of higher-order statistics and spectra”. In: *Stat. Sin.* 1 (1991), pp. 465–476.
- [10] Michael M Bronstein et al. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *arXiv preprint arXiv:2104.13478* (2021).
- [11] Joan Bruna and Stéphane Mallat. “Invariant scattering convolution networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1872–1886.
- [12] Joan Bruna and Stéphane Mallat. “Multiscale sparse microcanonical models”. In: *Mathematical Statistics and Learning* 1.3 (2019), pp. 257–315.

- [13] Terry M Caelli. “The prediction of interactions between visual forms by products of Lie operators”. In: *Mathematical Biosciences* 30.3-4 (1976), pp. 191–204.
- [14] Ho Yin Chau et al. “Disentangling images with Lie group transformations and sparse coding”. In: *arXiv preprint arXiv:2012.12071* (2020).
- [15] Ricky T. Q. Chen et al. *Isolating Sources of Disentanglement in Variational Autoencoders*. 2019. arXiv: 1802.04942 [cs.LG].
- [16] Xi Chen et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *arXiv preprint arXiv:1606.03657* (2016).
- [17] SueYeon Chung and LF Abbott. “Neural population geometry: An approach for understanding biological and artificial neural networks”. In: *arXiv preprint arXiv:2104.07059* (2021).
- [18] Taco Cohen and Max Welling. “Group equivariant convolutional networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 2990–2999.
- [19] Taco Cohen and Max Welling. “Learning the irreducible representations of commutative lie groups”. In: *International Conference on Machine Learning*. PMLR. 2014, pp. 1755–1763.
- [20] Taco Cohen et al. “Gauge equivariant convolutional networks and the icosahedral CNN”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1321–1330.
- [21] Benjamin J Culpepper and Bruno A Olshausen. “Learning transport operators for image manifolds.” In: *NIPS*. 2009, pp. 423–431.
- [22] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.
- [23] Jifeng Dai et al. “Deformable convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 764–773.
- [24] Nilesch Dalvi et al. “Adversarial classification”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 99–108.
- [25] Russell L De Valois, Duane G Albrecht, and Lisa G Thorell. “Spatial frequency selectivity of cells in macaque visual cortex”. In: *Vision research* 22.5 (1982), pp. 545–559.
- [26] Edgar A DeYoe et al. “Mapping striate and extrastriate visual areas in human cerebral cortex”. In: *Proceedings of the National Academy of Sciences* 93.6 (1996), pp. 2382–2386.

- [27] James J DiCarlo and David D Cox. “Untangling invariant object recognition”. In: *Trends in cognitive sciences* 11.8 (2007), pp. 333–341.
- [28] Charles J Duffy and Robert H Wurtz. “Sensitivity of MST neurons to optic flow stimuli. I. A continuum of response selectivity to large-field stimuli”. In: *Journal of neurophysiology* 65.6 (1991), pp. 1329–1345.
- [29] Charles J Duffy and Robert H Wurtz. “Sensitivity of MST neurons to optic flow stimuli. II. Mechanisms of response selectivity revealed by small-field stimuli”. In: *Journal of neurophysiology* 65.6 (1991), pp. 1346–1359.
- [30] Robert C Emerson et al. “Nonlinear directionally selective subunits in complex cells of cat striate cortex”. In: *Journal of Neurophysiology* 58.1 (1987), pp. 33–65.
- [31] Gidon Felsen et al. “Cortical sensitivity to visual features in natural scenes”. In: *PLoS biology* 3.10 (2005), e342.
- [32] Mario Ferraro and Terry M Caelli. “Lie transformation groups, integral transforms, and invariant pattern recognition.” In: *Spatial Vision* (1994).
- [33] Mario Ferraro and Terry M Caelli. “Relationship between integral transform invariances and Lie group theory”. In: *JOSA A* 5.5 (1988), pp. 738–742.
- [34] E Paxon Frady and Friedrich T Sommer. “Robust computation with rhythmic spike patterns”. In: *Proceedings of the National Academy of Sciences* 116.36 (2019), pp. 18050–18059.
- [35] Sheldon L Freud. *The physiological locus of the spiral after-effect*. Vol. 64. 17. Federal Aviation Agency, Office of Aviation Medicine, 1964.
- [36] K. Fukushima. “Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”. In: *Biological Cybernetics* 36 (1980), pp. 193–202.
- [37] Ken-Ichi Funahashi. “On the approximate realization of continuous mappings by neural networks”. In: *Neural networks* 2.3 (1989), pp. 183–192.
- [38] Jack L Gallant, Jochen Braun, and David C Van Essen. “Selectivity for polar, hyperbolic, and Cartesian gratings in macaque visual cortex”. In: *Science* 259.5091 (1993), pp. 100–103.
- [39] Juan A Gallego et al. “Neural manifolds for the control of movement”. In: *Neuron* 94.5 (2017), pp. 978–984.
- [40] Melvyn A Goodale and A David Milner. “Separate visual pathways for perception and action”. In: *Trends in neurosciences* 15.1 (1992), pp. 20–25.
- [41] Jonathan Green et al. “A neural circuit architecture for angular integration in *Drosophila*”. In: *Nature* 546.7656 (2017), pp. 101–106.

- [42] David B Grimes and Rajesh PN Rao. “Bilinear sparse coding for invariant vision”. In: *Neural computation* 17.1 (2005), pp. 47–73.
- [43] Charles G Gross, CE de Rocha-Miranda, and DB Bender. “Visual properties of neurons in inferotemporal cortex of the macaque.” In: *Journal of neurophysiology* 35.1 (1972), pp. 96–111.
- [44] Raia Hadsell, Sumit Chopra, and Yann LeCun. “Dimensionality reduction by learning an invariant mapping”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.
- [45] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.
- [46] MA Hietanen et al. “Phase sensitivity of complex cells in primary visual cortex”. In: *Neuroscience* 237 (2013), pp. 19–28.
- [47] Irina Higgins et al. “beta-vae: Learning basic visual concepts with a constrained variational framework”. In: (2016).
- [48] Geoffrey F Hinton. “A parallel computation that assigns canonical object-based frames of reference”. In: *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*. 1981, pp. 683–685.
- [49] William C Hoffman. “The Lie algebra of visual perception”. In: *Journal of mathematical Psychology* 3.1 (1966), pp. 65–98.
- [50] William C Hoffman. “The visual cortex is a contact bundle”. In: *Applied Mathematics and Computation* 32.2-3 (1989), pp. 137–167.
- [51] John J Hopfield. “Neural networks and physical systems with emergent collective computational abilities”. In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.
- [52] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.
- [53] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.
- [54] Qin Hu and Jonathan D Victor. “Two-dimensional hermite filters simplify the description of high-order statistics of natural images”. In: *Symmetry* 8.9 (2016), p. 98.

- [55] David H Hubel and Torsten N Wiesel. “Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat”. In: *Journal of neurophysiology* 28.2 (1965), pp. 229–289.
- [56] David H Hubel and Torsten N Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. In: *The Journal of physiology* 148.3 (1959), pp. 574–591.
- [57] David H Hubel and Torsten N Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of physiology* 160.1 (1962), pp. 106–154.
- [58] Claire V Hutchinson, Timothy Ledgeway, and Curtis L Baker. “Phase-dependent interactions in visual cortex to combinations of first-and second-order stimuli”. In: *Journal of Neuroscience* 36.49 (2016), pp. 12328–12337.
- [59] Aapo Hyvärinen. “Survey on independent component analysis”. In: (1999).
- [60] Jörn-Henrik Jacobsen et al. “Excessive invariance causes adversarial vulnerability”. In: *arXiv preprint arXiv:1811.00401* (2018).
- [61] Max Jaderberg et al. “Spatial transformer networks”. In: *arXiv preprint arXiv:1506.02025* (2015).
- [62] Ramakrishna Kakarala. “A group-theoretic approach to the triple correlation”. In: *[1993 Proceedings] IEEE Signal Processing Workshop on Higher-Order Statistics*. IEEE. 1993, pp. 28–32.
- [63] Ramakrishna Kakarala. “Completeness of bispectrum on compact groups”. In: *arXiv preprint arXiv:0902.0196* 1 (2009).
- [64] Ramakrishna Kakarala. “The bispectrum as a source of phase-sensitive invariants for Fourier descriptors: a group-theoretic approach”. In: *Journal of Mathematical Imaging and Vision* 44.3 (2012), pp. 341–353.
- [65] Ramakrishna Kakarala. “Triple correlation on groups”. PhD thesis. University of California, Irvine, 1992.
- [66] Hyunjik Kim and Andriy Mnih. “Disentangling by factorising”. In: *35th International Conference on Machine Learning, ICML 2018* 6 (2018), pp. 4153–4171.
- [67] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [68] R. Kondor. *Group theoretical methods in machine learning*. Columbia University, PhD Thesis, 2008.
- [69] Risi Kondor. “A novel set of rotationally and translationally invariant features for images based on the non-commutative bispectrum”. In: *arXiv preprint cs/0701127* (2007).

- [70] R Kondor et al. “Covariant compositional networks for learning graphs”. In: *arXiv preprint arXiv:1801.02144* (2018).
- [71] Gerhard Krieger et al. “Object and scene analysis by saccadic eye-movements: an investigation with higher-order statistics”. In: *Spatial vision* 13.2-3 (2000), pp. 201–214.
- [72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.
- [73] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015), pp. 1332–1338.
- [74] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.
- [75] Nikos K Logothetis, Jon Pauls, and Tomaso Poggio. “Shape representation in the inferior temporal cortex of monkeys”. In: *Current biology* 5.5 (1995), pp. 552–563.
- [76] Romain Lopez et al. “Information constraints on auto-encoding variational Bayes”. In: *Advances in Neural Information Processing Systems* 2018-Decem.1 (2018), pp. 6114–6125. ISSN: 10495258.
- [77] Christos Louizos et al. “The Variational Fair Autoencoder”. In: (2015), pp. 1–11. URL: <http://arxiv.org/abs/1511.00830>.
- [78] Donald M MacKay. “Moving visual images produced by regular stationary patterns”. In: *Nature* 180.4591 (1957), pp. 849–850.
- [79] Zoe F Mann and Matthew W Kelley. “Development of tonotopy in the auditory periphery”. In: *Hearing research* 276.1-2 (2011), pp. 2–15.
- [80] S Marçelja. “Mathematical description of the responses of simple cortical cells”. In: *JOSA* 70.11 (1980), pp. 1297–1300.
- [81] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.
- [82] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018).
- [83] Hamish Meffin et al. “Spatial phase sensitivity of complex cells in primary visual cortex depends on stimulus contrast”. In: *Journal of neurophysiology* 114.6 (2015), pp. 3326–3338.

- [84] Roland Memisevic and Geoffrey E Hinton. “Learning to represent spatial transformations with factored higher-order Boltzmann machines”. In: *Neural computation* 22.6 (2010), pp. 1473–1492.
- [85] Xu Miao and Rajesh PN Rao. “Learning the lie groups of visual invariance”. In: *Neural computation* 19.10 (2007), pp. 2665–2693.
- [86] Seyed-Mohsen Moosavi-Dezfooli et al. “Universal adversarial perturbations”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1765–1773.
- [87] Edvard I Moser, Emilio Kropff, and May-Britt Moser. “Place cells, grid cells, and the brain’s spatial representation system”. In: *Annu. Rev. Neurosci.* 31 (2008), pp. 69–89.
- [88] J Anthony Movshon, Ian D Thompson, and David J Tolhurst. “Receptive field organization of complex cells in the cat’s striate cortex.” In: *The Journal of physiology* 283.1 (1978), pp. 79–99.
- [89] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. “Pytorch metric learning”. In: *arXiv preprint arXiv:2008.09164* (2020).
- [90] Ivan Nazarov and Hendrick Schroter. *CplxModule*. <https://github.com/ivannz/cplxmodule>. 2021.
- [91] Chrysostomos L Nikias and Jerry M Mendel. “Signal processing with higher-order spectra”. In: *IEEE Signal processing magazine* 10.3 (1993), pp. 10–37.
- [92] André J Noest. “Discrete-state phasor neural networks”. In: *Physical Review A* 38.4 (1988), p. 2196.
- [93] Bruno A Olshausen, Charles H Anderson, and David C Van Essen. “A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information”. In: *Journal of Neuroscience* 13.11 (1993), pp. 4700–4719.
- [94] Bruno A Olshausen and David J Field. “Sparse coding with an overcomplete basis set: A strategy employed by V1?” In: *Vision research* 37.23 (1997), pp. 3311–3325.
- [95] Bruno A Olshausen et al. “Bilinear models of natural images”. In: *Human Vision and Electronic Imaging XII*. Vol. 6492. International Society for Optics and Photonics. 2007, p. 649206.
- [96] Alan V Oppenheim and Jae S Lim. “The importance of phase in signals”. In: *Proceedings of the IEEE* 69.5 (1981), pp. 529–541.
- [97] Guy A Orban et al. “First-order analysis of optical flow in monkey brain”. In: *Proceedings of the National Academy of Sciences* 89.7 (1992), pp. 2595–2599.

- [98] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.
- [99] Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572.
- [100] Walter Pitts and Warren S McCulloch. “How we know universals; the perception of auditory and visual forms”. en. In: *Bull. Math. Biophys.* 9.3 (Sept. 1947), pp. 127–147.
- [101] Henri Poincaré. *On the Foundations of Geometry*. Open Court Publishing Company, 1898.
- [102] Daniel A Pollen, Jame R Lee, and Joseph H Taylor. “How does the striate cortex begin the reconstruction of the visual world?” In: *Science* 173.3991 (1971), pp. 74–77.
- [103] William H Press et al. *Numerical recipes in Fortran 77: volume 1, volume 1 of Fortran numerical recipes: the art of scientific computing*. Cambridge university press, 1992.
- [104] Charles R Qi et al. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.
- [105] R Quian Quiroga et al. “Invariant visual representation by single neurons in the human brain”. In: *Nature* 435.7045 (2005), pp. 1102–1107.
- [106] Hubert Ramsauer et al. “Hopfield networks is all you need”. In: *arXiv preprint arXiv:2008.02217* (2020).
- [107] Rajesh PN Rao and Daniel L Ruderman. “Learning Lie groups for invariant visual perception”. In: *Advances in neural information processing systems* (1999), pp. 810–816.
- [108] Maximilian Riesenhuber and Tomaso Poggio. “Hierarchical models of object recognition in cortex”. In: *Nature neuroscience* 2.11 (1999), pp. 1019–1025.
- [109] ET Rolls and GC Baylis. “Size and contrast have only small effects on the responses to faces of neurons in the cortex of the superior temporal sulcus of the monkey”. In: *Experimental brain research* 65.1 (1986), pp. 38–48.
- [110] Sam T Roweis and Lawrence K Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *science* 290.5500 (2000), pp. 2323–2326.
- [111] Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 1962.
- [112] Nicole C Rust et al. “Spatiotemporal elements of macaque v1 receptive fields”. In: *Neuron* 46.6 (2005), pp. 945–956.

- [113] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. “Dynamic routing between capsules”. In: *arXiv preprint arXiv:1710.09829* (2017).
- [114] Hide-aki Saito et al. “Integration of direction signals of image motion in the superior temporal sulcus of the macaque monkey”. In: *Journal of Neuroscience* 6.1 (1986), pp. 145–157.
- [115] Sophia Sanborn et al. “Node2Motif: Hierarchical invariant embeddings of structured graphs using the bispectrum”. In: *NeurIPS Graph Representation Learning Workshop*. 2019.
- [116] H Sebastian Seung and Daniel D Lee. “The manifold ways of perception”. In: *science* 290.5500 (2000), pp. 2268–2269.
- [117] Roger N Shepard. “Perceptual-cognitive universals as reflections of the world”. In: *Psychonomic Bulletin & Review* 1.1 (1994), pp. 2–28.
- [118] Roger N Shepard and Jacqueline Metzler. “Mental rotation of three-dimensional objects”. In: *Science* 171.3972 (1971), pp. 701–703.
- [119] Bernt C Skottun. “A model for end-stopping in the visual cortex”. In: *Vision research* 38.13 (1998), pp. 2023–2035.
- [120] Jascha Sohl-Dickstein, Ching Ming Wang, and Bruno A Olshausen. “An unsupervised algorithm for learning lie group transformations”. In: *arXiv preprint arXiv:1001.1027* (2010).
- [121] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *arXiv preprint arXiv:1312.6199* (2013).
- [122] Keiji Tanaka, YOSHIRO Fukada, and HA Saito. “Underlying mechanisms of the response specificity of expansion/contraction and rotation cells in the dorsal part of the medial superior temporal area of the macaque monkey”. In: *Journal of neurophysiology* 62.3 (1989), pp. 642–656.
- [123] Keiji Tanaka and Hide-Aki Saito. “Analysis of motion of the visual field by direction, expansion/contraction, and rotation cells clustered in the dorsal part of the medial superior temporal area of the macaque monkey”. In: *Journal of neurophysiology* 62.3 (1989), pp. 626–641.
- [124] Keiji Tanaka et al. “Analysis of local and wide-field movements in the superior temporal visual areas of the macaque monkey”. In: *Journal of Neuroscience* 6.1 (1986), pp. 134–144.
- [125] Shiming Tang et al. “Complex pattern selectivity in macaque primary visual cortex revealed by large-scale two-photon imaging”. In: *Current Biology* 28.1 (2018), pp. 38–48.

- [126] Joshua B Tenenbaum, Vin De Silva, and John C Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.
- [127] Joshua B Tenenbaum and William T Freeman. “Separating style and content with bilinear models”. In: *Neural computation* 12.6 (2000), pp. 1247–1283.
- [128] Nathaniel Thomas et al. “Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds”. In: *arXiv preprint arXiv:1802.08219* (2018).
- [129] Jon Touryan, Gidon Felsen, and Yang Dan. “Spatial structure of complex cell receptive fields measured with natural images”. In: *Neuron* 45.5 (2005), pp. 781–791.
- [130] Jon Touryan, Brian Lau, and Yang Dan. “Isolation of relevant visual features from random stimuli for cortical complex cells”. In: *Journal of Neuroscience* 22.24 (2002), pp. 10811–10818.
- [131] Chiheb Trabelsi et al. “Deep complex networks”. In: *arXiv preprint arXiv:1705.09792* (2017).
- [132] J. Tukey. “The spectral representation and transformation properties of the higher moments of stationary time series”. In: *Reprinted in The Collected Works of John W. Tukey* 1 (1953), pp. 165–184.
- [133] DC Van Essen, JHR Maunsell, and JL Bixby. “The middle temporal visual area in the macaque: myeloarchitecture, connections, functional properties and topographic organization”. In: *Journal of Comparative Neurology* 199.3 (1981), pp. 293–326.
- [134] Luc Van Gool et al. “Vision and Lie’s approach to invariance”. In: *Image and vision computing* 13.4 (1995), pp. 259–277.
- [135] Jonathan D Victor et al. “Responses of V1 neurons to two-dimensional hermite functions”. In: *Journal of neurophysiology* 95.1 (2006), pp. 379–400.
- [136] Arash Yazdanbakhsh and Margaret S Livingstone. “End stopping in V1 is sensitive to contrast”. In: *Nature neuroscience* 9.5 (2006), pp. 697–702.
- [137] JI Yellott Jr and GJ Iverson. “Uniqueness theorems for generalized autocorrelations”. In: *Journal of the Optical Society of America* 9 (1992), pp. 388–404.
- [138] Molis Yunzab, Shaun L Cloherty, and Michael R Ibbotson. “Comparison of contrast-dependent phase sensitivity in primary visual cortex of mouse, cat and macaque”. In: *Neuroreport* 30.14 (2019), p. 960.

- [139] Christoph Zetsche and Florian Röhrbein. “Nonlinear and extra-classical receptive field properties and the statistics of natural scenes”. In: *Network: Computation in Neural Systems* 12.3 (2001), p. 331.
- [140] Junbo Zhao et al. “Stacked what-where auto-encoders”. In: *arXiv preprint arXiv:1506.02351* (2015).
- [141] Mengchen Zhu and Christopher J Rozell. “Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system”. In: *PLoS computational biology* 9.8 (2013), e1003191.