The Mechanics of CNN Filtering with Rectification

by

Liam FRIJA-ALTARAC

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN
PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS
M.A.Sc.

MONTREAL, DECEMBER 29TH 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Matthew Toews, Thesis director
Département de génie des systèmes, École de technologie supérieure

Mr. Rafael Menelau Oliveira Cruz, President of the board of examiners
Département de génie logiciel et des technologies de l'information, École de technologie supérieure

Mr. Marco Pedersoli, Member of the jury
Département de génie des systèmes, École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON DECEMBER 4TH 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Professor Matthew Toews, for this opportunity, which served as my introduction to the world of research. Our many discussions were invaluable and greatly expanded my perspective, far beyond what I had initially imagined within the field of machine learning. I am deeply grateful for his mentorship, guidance, and support throughout this work.

I would also like to thank my friends from the LIVIA Lab — Pranav Agarwal, Talia Vazquez Romaguera, Aryan Shukla, Muhammad Haseeb Aslam, and Muhammad Osama Zeeshan. Your friendship and support were greatly appreciated during this challenging period. I wish you continued success in your future endeavours.

Finally, I would like to thank my family — my mother and father, Katy and Remy (respectively), and my sister, Sharina. This thesis would not have been possible without your unwavering support and *incessant* encouragement. I dedicate this thesis to you.

La mécanique du filtrage dans les réseaux de neurones convolutifs avec rectification

Liam FRIJA-ALTARAC

RÉSUMÉ

Cette thèse constitue la première étude des propriétés mécaniques du filtrage convolutionnel avec rectification, étudiée sous l'angle des noyaux convolutionnels symétriques et antisymétriques. Les noyaux symétriques provoquent une diffusion isotrope du contenu de l'image sans déplacement net, tandis que les noyaux antisymétriques entraînent des effets de déplacement directionnel et d'atténuation lorsqu'ils sont convolués séquentiellement selon des orientations variables. La vitesse de déplacement de l'information est linéairement liée au rapport entre l'énergie antisymétrique et l'énergie totale du noyau. Les propriétés de symétrie sont analysées dans le domaine spectral à l'aide de la transformée en cosinus discrète (DCT), où la structure des petits filtres convolutionnels (par exemple 3×3 pixels) est dominée par les bases de basse fréquence, en particulier la composante continue DC Σ et les composantes de gradient ∇, qui définissent les modes fondamentaux de propagation de l'information.

En appliquant cette analyse aux filtres de CNN entraînés, nous constatons que les composantes fréquentielles d'ordre faible (notamment les bases DC et de gradient) dominent, représentant plus de 92% des performances de classification dans des modèles populaires tels que VGG16 et ResNet50. La symétrie des noyaux évolue avec la profondeur : les premières couches sont majoritairement antisymétriques, mettant en évidence les gradients orientés, tandis que les couches plus profondes deviennent progressivement plus symétriques, favorisant la diffusion. De plus, nous observons que les filtres s'organisent en structures d'orientation bipolaire corrélée à travers les canaux et les couches, maintenant une cohérence directionnelle entre les noyaux consécutifs tout en supprimant les activations orthogonales. Ce travail fournit un cadre systématique pour l'analyse, l'interprétation et, potentiellement, l'orientation de la conception d'architectures et d'algorithmes d'apprentissage améliorés à travers leur structure géométrique et spectrale.

Mots-clés: réseaux de neurones convolutifs (cnn), interprétabilité mécanistique, transformée en cosinus discrète (dct), analyse spectrale, visualisation de filtres, propagation de l'information, filtrage linéaire, rectification

The Mechanics of CNN Filtering with Rectification

Liam FRIJA-ALTARAC

ABSTRACT

This thesis is the first investigation of the mechanical properties of convolutional filtering with rectification through the lens of symmetric and antisymmetric convolutional kernels. Symmetric kernels cause image content to diffuse isotropically with no net displacement, whereas antisymmetric kernels cause directional displacement and attenuation effects when sequentially convolved at varying orientations. The speed of information displacement is linearly related to the ratio of antisymmetric vs total kernel energy. Symmetry properties are analyzed in the spectral domain via the discrete cosine transform (DCT), where the structure of small convolutional filters (e.g. 3×3 pixels) is dominated by low-frequency bases, specifically the DC Σ and gradient components ∇, which define the fundamental modes of information propagation.

Applying this analysis to trained CNN filters, we find that low-order frequency components (specifically DC and gradient bases) dominate, accounting for over 92% of classification performance in popular models such as VGG16 and ResNet50. The symmetry of kernels evolves with depth: early layers are predominantly antisymmetric, emphasizing oriented gradients, whereas deeper layers become increasingly symmetric, promoting diffusion. Furthermore, we observe that filters organize into correlated bipolar orientation structures across channels and layers, maintaining directional alignment between consecutive kernels while suppressing orthogonal activations. This work provides a systematic framework for analyzing, interpreting, and potentially guiding the design of improved architectures and learning algorithms through their geometric and spectral structure.

Keywords: convolutional neural networks (cnns), mechanistic interpretability, discrete cosine transform (dct), spectral analysis, filter visualization, information propagation, linear filtering, rectification

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
PCA	Principal Component Analysis
NMF	Non-negative Matrix Factorization
SGD	Stochastic Gradient Descent
AC	Alternating current
DC	Direct current
RF	Receptive Field
ERF	Effective Receptive Field
ReLU	Rectified Linear Unit

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

θ	Orientation in 2D image plane
f	Function
f_a	Antisymmetric function
f_s	Symmetric function
β^2	Squared ratio of antisymmetric energy to total energy $\frac{\|\|f_a\|\|^2}{\|\|f\|\|^2}$
Σ	DC (Average) operator
∇	Gradient operator
∇_x	2D gradient operator in horizontal direction
∇_y	2D gradient operator in vertical direction
∇_θ	2D gradient operator in orientation θ
$D_{u,v}$	A DCT basis function at frequency coordinates u, v
η	Learning Rate
$\hat{\theta}$	Dominant Orientation of a Filter in antisymmetric plane
ω	Weighting coefficient of DCT bases

INTRODUCTION

In recent decades, neural architectures (LeCun *et al.*, 2015) have seen a significant rise in popularity in both industry and research. Deep Neural Networks and Transformers (Vaswani *et al.*, 2017) have become the de facto state-of-the-art technologies for tasks in computer vision, natural language processing, and robotics (Sutton & Barto, 2018). Despite their widespread adoption, there is skepticism in several fields regarding their adoption. This lack of trust can be partly attributed to the "black box" nature of neural networks (Rudin, 2019; Zhang *et al.*, 2021). Due to their scale and the complexity of interactions between units, it remains challenging to understand how deep neural networks arrive at a given decision or how information flows through activation layers.

At the core of these models lies the rectified convolution operation, in which activations are transformed by linear filters and subsequently passed through nonlinear rectification. While convolution itself is linear and well understood, the systematic use of rectification poses a major obstacle to analysis using traditional linear tools such as spectral methods. Network interpretability has therefore become an active area of research aimed at addressing these challenges. In particular, mechanistic interpretability seeks to characterize the emergent structure and behaviour of learned convolutional filters at a low level, for instance through curvature detectors (Cammarata *et al.*, 2020) and feature visualizations (Voss *et al.*, 2021). Our work is aligned with this motivation in that we examine how convolutional operations propagate signals and how trained CNNs organize their kernels. Filter shapes have also been characterized using group theory, symmetry (Alsallakh *et al.*, 2025), and equivariance (Weiler *et al.*, 2018). However, to date, little is known about how symmetric and antisymmetric filter components transform activation images under rectified convolution.

This thesis proposes a unique model of the action of rectified convolution upon activation image information in CNN layers, in order to better understand and interpret the manner by which CNNs

route information from input to output. We develop a quantitative model of information flow that describes how signals propagate in image space through repeated sequences of convolution and rectification. A clearer understanding of these mechanisms helps illuminate phenomena that remain only partially understood, such as the expansion of the effective receptive field (ERF) observed in trained networks relative to random initialization (Luo *et al.*, 2016). We hypothesize that such insights can inform principled architectural design, guide inductive biases for learning, and contribute to the development of more efficient and mechanistically interpretable convolutional models.

Our model focuses on individual kernels, which are decomposed into unique symmetric and antisymmetric components (also known as even-odd components). When applied to convolutional filters, this decomposition reveals two distinct modes of information propagation. Symmetric components give rise to diffusive behaviour that preserves the centre of mass of activations. Antisymmetric components induce directional translation of activation energy. This dichotomy allows rectified convolution to be interpreted as a mixture of diffusion and drift processes in image space, analogously to potential and kinetic energy in physical systems, as well as diffusion-advection differential equations.

Our model shows that for small discrete filters typically used in CNNs (eg. 3×3), the symmetric and antisymmetric components are dominated by primary sum (Σ) and gradient (∇) operators, respectively. These correspond to low-frequency components under a spectral decomposition such as the discrete cosine transform. Under convolution with ReLU, this decomposition allows us to interpret information flow as a mixture of symmetric diffusion and oriented scattering processes, respectively. Symmetric filtering does not change the centre of mass of activations, while antisymmetric filtering with rectification causes the centre of mass to shift, similarly to kinetic energy. This translational shift may be modelled by a binomial pyramid with ReLU and manifests as vibration or unidirectional motion in a direction perpendicular to the gradient

of structures in the original image. While our analysis focuses on individual 2D kernels, it generalizes to higher dimensional spaces (*i.e*. 3D images) via symmetry and multiple channels via linear superposition.

The main novel findings of this thesis are as follows:

- Symmetric kernels preserve the centre of mass of activation patterns, inducing isotropic diffusion, while antisymmetric kernels displace the centre of mass, producing momentum-like propagation of activation image information.
- Convolution with rectification leads to maximal propagation distance compared to convolution without rectification; we offer a proof and demonstration.
- Sequential convolution at varying orientations causes attenuation of activation energy, with maximal attenuation occurring when kernel orientations differ by $\pi/2$ rad.
- CNNs trained on ImageNet are dominated by low-order DCT components, specifically the DC (Σ) and orientation gradient (∇) bases, which together are responsible for over 92% of accuracy.
- Early-layer filters are predominantly antisymmetric and orientation selective, whereas deeper layers exhibit increasing symmetry, suggesting a transition from local feature extraction to global integration.
- Filters throughout the network develop a correlated bipolar structure along antisymmetric orientations, where channels within a filter tend to share a common or opposite dominant direction defined by ∇.
- Oriented kernels operate preferentially on similarly oriented filters from preceding layers, while suppressing activations from orthogonally oriented filters, suggesting a systematic translation of image content across the entire depth of the CNN.

The remainder of this thesis is organized as follows. Chapter 1 presents a short survey of the history of CNNs, interpretability and the study of learned filters through spectral methods.

Chapter 2 defines the antisymmetric framework which we use throughout the project. We describe the convolution properties of symmetric and antisymmetric operators under ReLU. Chapter 3 analyzes the geometry of learned CNN kernels for VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016). We first validate their frequency composition and demonstrate that energy in trained networks resides mostly in low frequencies. We then examine the emergent geometric properties of the kernels and find an orientation relationship between filter-channels which extends across depth, where oriented kernels operate preferentially on similarly oriented filters from preceding layers.

CHAPTER 1

LITERATURE REVIEW

This thesis proposes a novel mathematical framework for understanding and interpreting the mechanics by which information flows in deep Convolutional Neural Networks with rectification, which are often treated as black boxes. The related literature thus pertains to convolutional neural networks, interpretation methods and signal processing properties. The majority of prior signal processing work has not considered rectified convolution, which has become standard in modern deep neural networks (Nair & Hinton, 2010), therein lies the contribution of this thesis.

1.1 Convolutional Neural Networks

Neural networks were invented in the 1950s as a computational analogy of neural processing, where individual neurons multiply inputs by weights and accumulate the result, for example the Multi-Layer Perceptron (MLP) (Minsky & Papert, 1988). Major developments included training via the backpropagation algorithm (Rumelhart *et al.*, 1985), automatically training neural network parameters without having to specify the computational form of weights by hand. Convolutional neural networks (CNNs) were invented in the 1980s (Fukushima, 2013; LeCun *et al.*, 1989), in order to reduce the number of parameters of the MLP from fully connected networks to small sets of shared weights shared across all image locations, known in signal processing as filters. Filters are capable of representing and making use of local image information, *i.e.* interactions between pixels located in close proximity within the image.

Modern deep learning began in 2012, when CNNs were trained using highly parallelized processing on a graphics processing unit (GPU) (Krizhevsky *et al.*, 2009), achieving a significant increase in performance on the benchmark ImageNet dataset. This allowed the development of various CNN architectures such as VGG (Simonyan & Zisserman, 2015) and ResNet (He *et al.*, 2016). The success of CNN models included the use of ReLU non-linearity (Maas *et al.*, 2013; Nair & Hinton, 2010), correct weight initialization (Glorot & Bengio, 2010; He *et al.*, 2015) and the use of small filter sizes, *i.e.* 3×3 pixel sizes. ReLU is a major component of modern neural

architectures, having replaced the sigmoid activation function that was widely used in earlier networks. Its main advantage lies in reducing the vanishing gradient problem that often arises with saturating nonlinearities such as the sigmoid (Glorot *et al.*, 2011).

A primary novelty of this work is in analyzing filtering with rectification, a non-linear operation. Linear filtering, including binomial filtering (Aubury & Luk, 1996), wavelets (Gabor, 1946; Haar, 1909) have been studied for over a century, whereas rectified filtering has only recently become ubiquitous in Deep Neural Networks. We find distinct properties when analyzed in terms of symmetric and antisymmetric filter components.

More recent developments include the so-called Transformer architecture (Vaswani *et al.*, 2017) which uses an attention mechanism to model non-local information and relationships between image patterns separated in the image, however computation remains focused on dot product and rectification operations.

Our work investigates specifically the geometry and mechanical action of filters, particularly radial symmetric and antisymmetric components of small filters. A variety of deep convolutional neural network architectures have been proposed. An interesting trend has been the use of low-resolution filters of odd square dimensions, most notably 3×3. Intuitively, low-resolution filters allow a larger number of channels and improved classification, 3×3 being the most popular choice for 2D CNNs. Even dimensions such as 2×2 are avoided, as the result of convolution cannot be stored symmetrically with respect to the original data, leading to an undesirable shift (Wu *et al.*, 2019). Nevertheless, Wu *et al.* (2019) show that this issue can be mitigated through appropriate padding strategies, achieving comparable accuracies to 3×3 models while using fewer parameters. Given the use of small filters, one might reasonably assume that useful aspects of information propagation are dominated by discrete effects, for example, primary lowest frequency components of the frequency spectrum. Our work presents a novel result that indeed, CNN accuracy is dominated by the average and gradient spectral components, which may be generalized to arbitrary filter sizes through radial symmetry and antisymmetry.

1.2 Interpretability

Our work falls within the scope of passive (*post-hoc*) interpretability methods, in which the goal is to explain and better understand the mechanisms that have been learned by a neural network (Zhang *et al.*, 2021). In passive interpretability, most existing works fall under the category of Local Attribution (Zhang *et al.*, 2021). The goal of such methods is to visualize salient features that contribute the most in a given task (Selvaraju *et al.*, 2017; Zeiler & Fergus, 2014). Local Attribution methods have moderate explanatory power (Zhang *et al.*, 2021). While they are capable of explaining which areas of an image were most "looked at" by the network, understanding *why* the network considered said area to be salient remains unknown. Rule based interpretability aims to answer such questions. Typically, rule based papers such as Fu (1991), aim to explain the inner workings of the network through logical expressions. Our work lies closer to rule based interpretability and mechanistic interpretability (Bereska & Gavves, 2024; Olah *et al.*, 2020b). However, our work is less about identifying semantic rules; we seek rather to understand how these rules are implemented (after backpropagation) in CNNs through the limited mechanisms that a 3×3 kernel can have. Many other works in machine learning propose various methods to analyze neural network, and provide certain observation that might further our general understanding of them. This is our goal for this thesis.

Similar works have investigated the learned mechanisms and properties of CNNs. For example, Luo *et al.* (2016) examined how the Effective Receptive Field (ERF) evolves during training. They empirically and analytically observed that the ERF of a randomly initialized CNN grows linearly with depth n at a rate of $O(\sqrt{n})$, though, relative to the theoretical receptive field, it shrinks at a rate of $O(1/\sqrt{n})$. After training, however, Luo *et al.* (2016) observed that the ERF typically grows to match the theoretical receptive field, though the underlying cause remains unclear. Our work shows that the size of the ERF is determined by the antisymmetric component. Our findings show that the learned oriented geometry in the antisymmetric component causes directional translation and is responsible for the growth of the ERF post-training.

Developments in computer vision have been inspired by biological neural networks, for example, the work of Hubel & Wiesel (1962) which famously described the mammalian optical receptive fields in early vision as being responsive to oriented edges. Following Hubel & Wiesel (1962), image gradient orientation became a central theme in computer vision, notably in the Scale-Invariant Feature Transform (SIFT) method (Lowe, 1999) and the Scattering Transform (Bruna & Mallat, 2013).

Same papers have looked into interpreting deep CNN layers. Studies by Cammarata *et al.* (2020, 2021); Olah *et al.* (2018, 2020a,b,c); Petrov *et al.* (2021) have looked in depth into the taxonomies of different learned filters in deep CNNs. Using Activation Maximization (Erhan *et al.*, 2009; Simonyan *et al.*, 2013) to visualize the features, Olah *et al.* (2018) demonstrate how features of an earlier layer are then used by subsequent layers to build a more complex basis. This finding is in line with the research done by Bau *et al.* (2020) who demonstrate that deep neural networks learn a "disentangled representation", meaning that a set of deep filters will have responses correlated with a certain class or pattern. This is in contrast to a "distributed representation" (Hinton *et al.*, 1986) which hypothesizes that a representation for a given class is encoded over multiple neurons. Decoding this representation would require understanding the learned mathematical relation, however, Bau *et al.* (2020) find that deep CNNs learn representations in which individual neurons can be semantically associated to a class (*e.g.* dog filters, flower filters), thus simplifying the task of interpretability. Olah *et al.* (2018, 2020a) find that a semantic attribution can also be done in early layers for lower level features, such as basic textures (*e.g.* lines, curves).

The spirit of our work and findings is closely related to those of Cammarata *et al.* (2020, 2021); Olah *et al.* (2018, 2020a,b,c); Petrov *et al.* (2021) who provide observations regarding the effect of various filter shapes, and visualization methods for filters and activations (Erhan *et al.*, 2009; Simonyan *et al.*, 2013). For example, Petrov *et al.* (2021) observed that oriented filter structure or *weight banding* emerges in deep layers, and that the orientation changes with training data rotation (*i.e.* changing from vertical to horizontal with a $\frac{\pi}{2}$ *rads* rotation), indicating that filter orientation reflects training image gradient. Our proposed framework allows for a simplified

visualization of the filter kernels in a space defined by symmetric and antisymmetric energy axes, thus allowing for an intuitive low-level understanding of the learned mechanics. For example, bipolar structures have been observed in deep filters, Cammarata *et al.* (2020) find that filters maximized for a given orientation θ also respond to stimuli rotated by π. This result is consistent with research on early filters in the mammalian visual system (Hubel & Wiesel, 1962). Voss *et al.* (2021), observe *orbital structures* in expanded weights (approximate cumulative effect of a filter at a given depth). Our work and visualization allow for an intuitive understanding of the bipolar orientation structure and symmetric energy structure of individual kernels and filters.

In addition to the qualitative analysis of filter geometry, recent work has studied symmetry in CNN kernels more formally. Alsallakh *et al.* (2025) examined the learned geometry of CNN filters from the perspective of rotational symmetry. We extend their findings by also quantifying the complementary antisymmetric component, which we parameterize by its orientation and magnitude. While our findings on symmetry are similar, rather than averaging all kernels to assess overall symmetry, we analyze both the symmetric and antisymmetric energy at the individual kernel level.

1.3 Spectral Analysis and Frequency Representations

Spectral methods (Fourier, DCT, PCA, etc.) reveal structure in images and learned model parameters. Natural images have scale-invariant, heavy-tailed spectra. Ruderman & Bialek (1993) showed that the average power spectrum of natural scenes follows a $1/f^{\alpha}$ law, meaning that most natural image energy is concentrated in low frequencies. Image compression algorithms such as JPEG (Wallace, 1991) exploit this by computing 8×8 Discrete Cosine Transforms (DCT) (Ahmed *et al.*, 1974) and discarding high-frequency coefficients following a *zig-zag* scanning pattern. In CNNs, spectral methods can be used to reparameterize or compress filters and reveal underlying structure in learned weights. Early work such as Bell & Sejnowski (1996) used principal component analysis (PCA) and Independent component analysis (ICA) on image patches to discover edge and contour-like basis filters, suggesting that CNNs might learn similar

frequency-distributed filters. Below, we review the role of various spectral transforms (DCT, Fourier, PCA) in how they have been used to study or shape learned CNN parameters.

The Discrete Fourier Transform (DFT) represents a signal as a sum of complex exponentials, sampled at integer frequency multiples, as shown in Equation (1.1).

$$X_{u,v} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F_{x,y} \cdot e^{-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{1.1}$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F_{x,y} \left[\cos\left(2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right) - i \sin\left(2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right)\right] \tag{1.2}$$

For a real-valued image ($F(x, y) \in \mathbb{R}^{M \times N}$), the DFT ($X_{u,v} \in \mathbb{C}^{M \times N}$) yields complex coefficients with redundant values. The DFT is cyclical-shift invariant, meaning that the DFT bases effectively treat boundary pixels as if they are adjacent. In practice, images are rarely periodic, and treating boundary pixels as neighbours introduces discontinuities that manifest as high-frequency *spikes*, requiring many high-order coefficients to represent.

In contrast, the Discrete Cosine Transform (DCT) (eq. (1.3)) uses only cosine (real, even) basis functions sampled at half-integer frequency multiples. This allows both periodic and antiperiodic bases, and thus is able to represent boundary discontinuities in natural images with a smaller number of low frequency components.

$$X_{u,v} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F_{x,y} \cos\left[\frac{\pi(2x+1)u}{2M}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{1.3}$$

Equation (1.3) computes the weight $X_{u,v} \in R^1$ of the cosine basis component defined over horizontal and vertical frequency (u, v) as the projection (dot-product) of the image $F_{x,y}$ defined over (x, y) coordinates. When looking individually at the bases formed by a single cosine in either the vertical or horizontal direction (*i.e.* when either $u = 0$ or $v = 0$), we can express them as fundamental bases, meaning that they cannot be expressed as a product of 2 other bases.

In Figure 1.1, these fundamental bases are noted by Σ, ∇, ∇^2 in the topmost row and leftmost column.

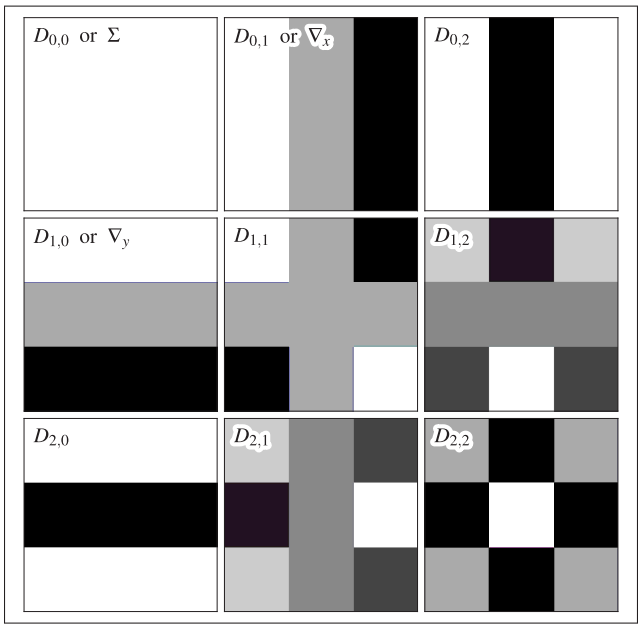


Figure 1.1 Nomenclature we use for 3×3 DCT bases

DCT coefficients are real-valued, making interpretation easier; each DCT basis image is a cosine-wave pattern of a certain frequency and orientation. By contrast, DFT coefficients come in complex conjugate pairs (sine and cosine at each frequency), complicating analysis. Thus, for visualization and interpretability, DCTs are often preferable as they directly reveal which spatial frequencies (*e.g.* horizontal and vertical edges) a filter captures. We therefore focus on DCT, while noting that some methods use full Fourier-based representations (*e.g.* for optimization speed (Rippel *et al.*, 2015)) but yield little insight into filter geometry.

One practical use of DCT in CNNs is for filter parameterization. Instead of learning each filter weight in pixel space, the DCT coefficients can be learned (through backpropagation), then inverted to get the spatial kernel. Chęiński & Wawrzyński (2020) propose DCT-Conv layers where each convolution filter is generated by taking an inverse DCT of a small set of trained spectral coefficients. They find that in training, many high-frequency DCT terms are near zero, so most coefficients can be frozen without hurting the accuracy.

Spectral analysis can provide a lens for understanding certain learned properties of CNN filters. Alsallakh *et al.* (2025) investigate the average learned geometry of CNN kernels through the lens of rotational symmetry and find that they tend to be highly symmetric about the center indicating that the network has implicitly preferred low-frequency (Σ) structured filter.

Besides DCT, matrix factorization methods (PCA, NMF) can be used to interpret weights with similar results. For example, Fukuzaki & Ikehara (2022) use Principal Component Analysis (PCA) to examen the principal components of VGG16 (Simonyan & Zisserman, 2015) across layers. Their results show a concentration of energy in low order components, and as found by Bell & Sejnowski (1996), the principal components of the learned kernels bear a strong resemblance to the DCT DC and gradient bases.

From a signal processing standpoint, ReLU acts as a classic half-wave rectifier. The spectral effects of ReLU have been studied. Kechris *et al.* (2024) derive a Fourier-domain description of ReLU, showing that it preserves the input's original frequencies while additionally introducing higher-frequency components and a constant DC offset. This is consistent with the classical analog; rectifying a sinusoid yields a doubled frequency and a DC shift.

In summary, prior work has separately addressed spectral representations and kernel geometry. The following chapter integrates these perspectives by formulating a symmetry-antisymmetry-based description of convolutional propagation under rectification.

CHAPTER 2

FILTER SYMMETRY AND THE MECHANICS OF RECTIFIED CONVOLUTION

Information in deep convolutional neural networks (CNNs) flows from the input image to the output due to the action of sequential convolution filtering, ReLU and subsampling layers. However, as convolution filter parameters are automatically derived from the backpropagation algorithm and its variants, little is known as to how filters route information in image space; the CNN is often regarded as a black box (Rudin, 2019). This chapter proposes to study filters in terms of their geometry and the action they have on input activation data in combination with the non-linear ReLU operation.

This section presents a novel framework for characterizing filters geometrically in terms of symmetric and antisymmetric components, which in convolution followed by rectification cause information to diffuse symmetrically or propagate directionally. The framework is first presented in the minimal case of 1D image space and a single filter channel, where sum $\Sigma = [1, 1]$ and difference $\nabla = [-1, 1]$ operators lead to diffusion and directional left-right motion that may be characterized by the binomial pyramid. The theory is then generalized to 2D image space and larger filter sizes via symmetry and to multi-channel filters via superposition, where antisymmetric directional components are further characterized by an in-plane angle θ.

2.1 Binomial Filtering with ReLU

The primary aspects of our theory are defined in this section in 1D image space in terms of binomial filtering, and then generalized to 2D image space and multiple channels using geometrical symmetry and superposition.

CNNs typically apply convolution followed by rectification using the so-called Rectified Linear Unit (ReLU) function defined as:

$$ReLU(I) = \max(0, I) = \frac{I + |I|}{2}, \tag{2.1}$$

where in Equation (2.1), $I \in R$ represents an individual image intensity. The activation image $I^{(t)}$ a time t is the result of convolution $*$ with a operator or filter F followed by ReLU:

$$I^{(t)} = ReLU(I^{(t-1)} * F), \tag{2.2}$$

Consider the minimal example of a 1×2 filter or operator F that may take on one of three values $F \in \{\Sigma, \nabla_-, \nabla_+\}$, where $\Sigma = [1, 1]$ is the sum operator, and $\nabla_- = [1, -1]$ and $\nabla_+ = [-1, 1]$ represent left-handed and right-handed gradient operators (note that $\nabla_- = -\nabla_+$). Furthermore, let $I^{(0)} = [0, 1, 0]$ represent an impulse in a 1D activation image of size 1x3 pixels, with a single non-zero pixel $I^{(0)}[0] = 1$ at centre position $x = 0$ at time (or layer) $t = 0$.

Inspection reveals that convolving image $I^{(0)}$ with these operators followed by ReLU leads to three distinct transformations of the original image information, *i.e*. symmetric diffusion and directional shifting to the left or right:

$$\begin{aligned} ReLU([0, 1, 0] * \Sigma) &= [0, 1, 1, 0], && \text{Diffusion} \\ ReLU([0, 1, 0] * \nabla_-) &= [0, 1, 0, 0], && \text{Shift Left} \\ ReLU([0, 1, 0] * \nabla_+) &= [0, 0, 1, 0], && \text{Shift Right} \end{aligned} \tag{2.3}$$

Repeated convolution of a pixel impulse with the sum operator $\Sigma = [1, 1]$ leads to the well-known binomial pyramid as shown in Figure 2.1, which may be viewed as a random walk where the value of $I^{(t)}(x)$ represents the number of possible paths leading to position x. Note that rectification has no effect if activations and filter coefficients are all positive. If activations are normalized to sum to unit length $|I^{(t)}| = 1$, then the image approaches a Normal distribution with standard deviation $\sigma = \sqrt{t}$, *i.e*. as $t \to \infty$, $I^{(t)} \to Normal(\sqrt{t})$.

The sum operator is symmetric and causes information to propagate isotropically in all directions. While the leading edge of diffusion propagates at maximum velocity, the binomial theorem states that the effective receptive field after t layers approximates a Gaussian distribution with

$$
\begin{aligned}
I^{(0)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(1)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(2)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 2 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(3)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 3 & 3 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(4)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 4 & 6 & 4 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(5)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 5 & 10 & 10 & 5 & \mathbf{1} & 0 & 0 \end{bmatrix}.
\end{aligned}
$$

Figure 2.1 Diffusion steps in symmetric 1-D convolution $I^{t+1} = \mathrm{ReLU}(I^t * [1, 1])$ lead to the well-known binomial pyramid

standard deviation $\sigma = \sqrt{t}$, as has been experimentally verified (Luo *et al.*, 2016). Diffusion has been studied through the Gaussian scale-space (Lindeberg, 1994), and in currently popular diffusion generative models (Sohl-Dickstein *et al.*, 2015).

Repeated convolution and rectification of an impulse with the gradient operators $F \in \{\nabla_-, \nabla_+\}$ causes information to follow unique paths in a random walk, as shown in the following illustrations. Figure 2.2 shows the effect of sequential convolution with exclusively either ∇_- or ∇_+ operator, which causes information to translate at a maximum constant speed in a single direction, *i.e.* either right or left as determined by the operator. Figure 2.3 show how alternately convolving with ∇_- and ∇_+ leads to a vibration mode. The probability of a specific path may be defined by the binomial distribution, where, assuming equiprobable left and right operators, the translation path is the least probable outcome, and vibration (Figure 2.3) represents the most probable outcome.

The vibration and translation modes observed here represent a departure from standard linear signal processing and result from the convolution of an input with odd filters followed by ReLU non-linearity. Vibration follows from convolutions alternating between left and right gradient filters, and produces no net motion of information within the image. Translation follows from convolutions with constant left or right gradient filters, and results in information propagating at the maximum speed, here $c = 0.5$ pixels per layer. Such vibration and translation modes can

$$
\begin{aligned}
I^{(0)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(2)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(3)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}, \\
I^{(4)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}.
\end{aligned}
$$

Figure 2.2 Translation steps in antisymmetric 1-D convolution, $I^{t+1} = ReLU(I^t * [-1, 1])$

$$
\begin{aligned}
I^{(0)} &= \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 \end{bmatrix}, \\
I^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \end{bmatrix}, \\
I^{(2)} &= \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \end{bmatrix}, \\
I^{(3)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \end{bmatrix}, \\
I^{(4)} &= \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned}
$$

Figure 2.3 Vibration steps in antisymmetric 1-D convolution,

$$
I^{t+1} = \begin{cases} ReLU(I^t * [-1, 1]) & \text{t } \textit{is even} \\ ReLU(I^t * [1, -1]) & \text{t } \textit{is odd} \end{cases}
$$

be expected in wavelet scattering models Bruna & Mallat (2013). In the following section, we generalize these modes to larger filters and 2D space via symmetry.

2.2 Filter Symmetry and Antisymmetry

The previous section presents our theory of information propagation in the minimal context, where the $\Sigma = [1, 1]$ and $\nabla = [-1, 1]$ operators completely span the space of $1x2$-pixel filters in a single spatial dimension. Here we show how these operators may be generalized to larger filter sizes (i.e. greater than 2 pixels) and higher image dimensions (i.e. 2D) using rotational symmetry and antisymmetry, where Σ and ∇ become elements of larger symmetric and antisymmetric filter sets.

The symmetric/antisymmetric decomposition provides a basic taxonomy for understanding how kernels propagate information through convolution. Symmetric kernels have an isotropic effect on image content, *e.g.* smoothing, local averaging and reducing high-frequency components. Antisymmetric kernels (*e.g.* finite-difference, Sobel, or derivative filters) tend to translate or emphasize directional changes in the signal, highlighting edges, gradients, or movement.

While filter symmetry has been studied (Alsallakh *et al.*, 2025), our work here is the first to define the mechanical effect of filter symmetry under rectified convolution. We show symmetric kernels cause isotropic diffusion while preserving the image center of mass, whereas antisymmetric kernels cause a displacement in the center of mass, and furthermore, the speed of this displacement is determined by the ratio of antisymmetric to total filter energy.

2.2.1 Symmetry in 1D Filters

Any function $f(x)$ may be decomposed as the binary sum of orthogonal symmetric (even) $f_s(x)$ and antisymmetric (odd) $f_a(x)$ components :

$$f(x) = f_s(x) + f_a(x), \tag{2.4}$$

where $f_s(x)$ and $f_a(x)$ are referred to as the symmetric and antisymmetric components. In 1D image space $x \in R^1$, these are defined in terms of $f(x)$ as follows:

$$\begin{aligned} f_s(x) &= \frac{f(x) + f(-x)}{2}, \\ f_a(x) &= \frac{f(x) - f(-x)}{2} = f(x) - f_s(x). \end{aligned} \tag{2.5}$$

The squared magnitude of a function $f(x)$ is defined as follows:

$$\|f(x)\|^2 = \sum_x f^2(x), \tag{2.6}$$

The squared magnitude may also be expressed as the squared magnitude of symmetric and antisymmetric components.

$$\|f(x)\|^2 = \|f_s(x)\|^2 + \|f_a(x)\|^2 \tag{2.7}$$

See Figure 2.4 for a minimal 1D example of a discrete function $f(x)$ decomposed into its symmetric-antisymmetric components.

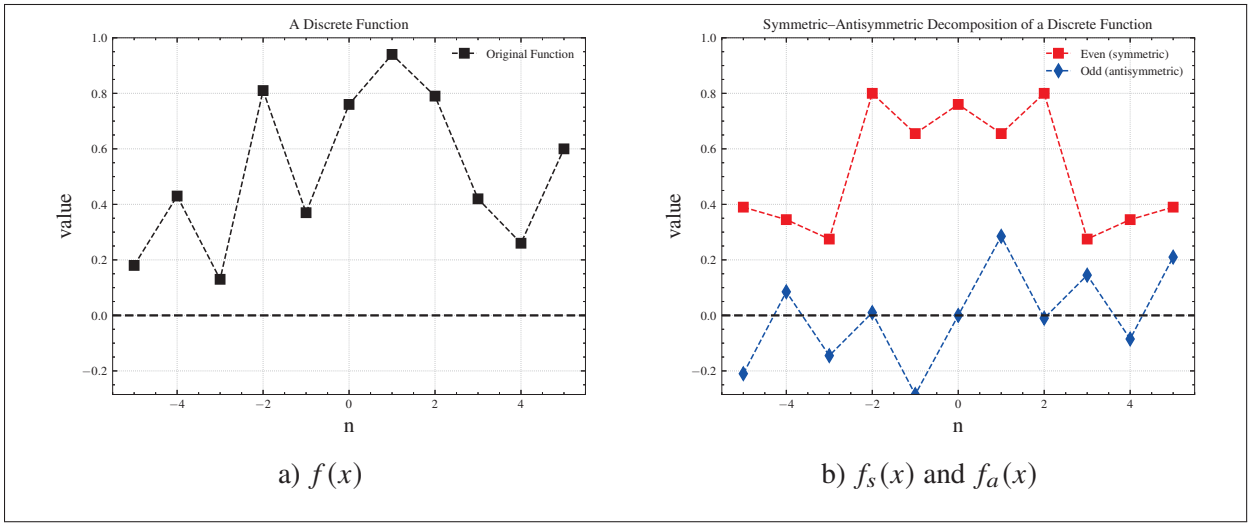


Figure 2.4 a) An example 1D function $f(x)$ and b) its decomposition into symmetric $f_s(x)$ (red) and antisymmetric $f_a(x)$ components

Finally, when examining an image or kernel f, decomposed into antisymmetric f_a and symmetric f_s components, we define a ratio β^2 to quantify the antisymmetric contribution to the total kernel energy as the ratio of the squared antisymmetric energy $||f_a||^2$ to the total squared filter energy $||f||^2$ as follows:

$$\beta^2 = \frac{||f_a||^2}{||f_a||^2 + ||f_s||^2}. \tag{2.8}$$

For example, $\beta^2 = 1$ would describe a purely antisymmetric (*e.g.* Sobel, Prewitt) kernel, and $\beta = 0$ would describe a rotationally symmetric (*e.g.* Gaussian, Laplacian, Averaging) kernel.

When $\hat{f}_a = \frac{f_a}{\|f_a\|}$ and $\hat{f}_s = \frac{f_s}{\|f_s\|}$ are unit vectors, a filter f may be defined according to the following mixing equation:

$$f = f_a + f_s = ||f||\beta\hat{f}_a + ||f||\sqrt{1-\beta^2}\hat{f}_s \tag{2.9}$$

Equation (2.9) allows us to generate kernels f of magnitude $||f||$ given arbitrary forms of f_a and f_s and a mixing ratio β in demonstrations and experiments.

We show in the following section how, under rectified convolution, the β^2 ratio determines the speed of information displacement, which ranges from 0 to a maximum speed determined by the filter size.

2.2.2 Symmetry in 2D Filters

Symmetry may be generalized to 2D images and filters as follows. Let $f(x, y) \in \mathbb{R}^1$ be a 2D function (e.g. a discrete kernel or an image) or mapping $f : \mathbb{Z}^2 \to \mathbb{R}^1$ from discrete image coordinates $(x, y) \in \mathbb{Z}^2_{>0}$ to a scalar value $f \in \mathbb{R}^1$. As in the case of 1D functions, a 2D function $f(x, y)$ may be decomposed into a sum $f(x, y) = f_s(x, y) + f_a(x, y)$ of orthogonal symmetric $f_s(x, y)$ and anti-symmetric $f_a(x, y)$ components, whose magnitudes of components follow a Pythagorean relationship as shown in Figure 2.5.

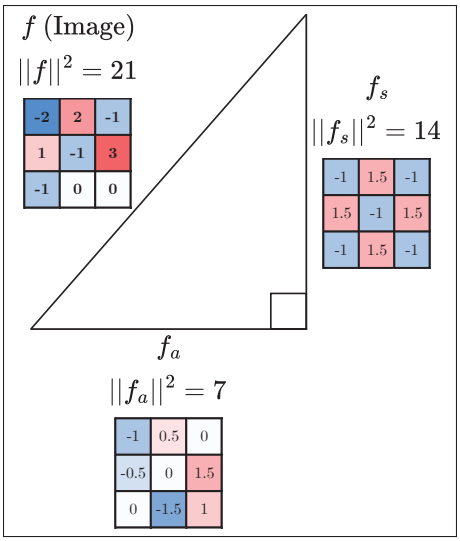


Figure 2.5 An example illustrating the Pythagorean geometry of a 2D function f (discrete 3×3 kernel) decomposed into orthogonal antisymmetric f_a and symmetric f_s components

The primary properties of symmetric and antisymmetric functions in 2D are as follows.

Definition 2.1 (Symmetric Image)**.** *An image $f_s(x, y) \in \mathbb{R}^{n \times n}$ is rotationally symmetric if*

$$f_s(x, y) = f_s(\pm x, \pm y) = f_s(\pm y, \pm x),$$

where the unique value of $f_s(x, y)$ is the average of the set of equidistant points $\{(x, y) : r = \sqrt{x^2 + y^2}\}$ forming sign and coordinate permutations of (x, y):

$$f_s(x, y) = \frac{1}{8} \sum_{s_x, s_y \in \{\pm 1\}} f(s_x x, s_y y) + f(s_y y, s_x x) \tag{2.10}$$

Definition 2.2 (Anti-Symmetric Image)**.** *An image $f_a(x, y) \in \mathbb{R}^{n \times n}$ is rotationally anti-symmetric if*

$$f_a(x, y) = f(x, y) - f_s(x, y)$$

where the sum of $f_a(x, y)$ over the set of equidistant points $\{(x, y) : r = \sqrt{x^2 + y^2}\}$ forming sign and coordinate permutations of (x, y) is 0:

$$\sum_{s_x, s_y \in \{\pm 1\}} f_a(s_x x, s_y y) + f_a(s_y y, s_x x) = 0 \tag{2.11}$$

Lemma 2.1 (Orthogonality)**.** *Symmetric and antisymmetric components are orthogonal and their scalar or dot product is thus 0:*

$$f_s(x, y) \cdot f_a(x, y) = \sum_{x,y} f_s(x, y) f_a(x, y) = 0$$

Definition 2.3 (Filter Energy)**.** *The energy of a filter f is defined as the squared magnitude $\|f(x, y)\|^2$, which equals the sum of squared magnitudes of the symmetric and antisymmetric components:*

$$\|f(x, y)\|^2 = \sum_{x,y} f^2(x, y) = \sum_{x,y} f_s^2(x, y) + \sum_{x,y} f_a^2(x, y).$$

2.2.3 Mechanical Properties of Rectified Convolution

Theorem 2.2 (Symmetric convolution preserves centre of mass.)**.** *The expected value $E_f[x, y]$ or centre of mass of a rectified activation image $f(x, y) \in [0, +\infty]$ does not change following convolution and rectification with a symmetric filter $f_s(x, y)$.*

$$E_f[x, y] = \frac{\sum_{x,y}[x, y] f(x, y)}{\sum_{x,y} f(x, y)} = \frac{\sum_{x,y}[x, y] ReLU\left[f(x, y) * f_s(x, y)\right]}{\sum_{x,y} ReLU\left[f(x, y) * f_s(x, y)\right]}$$

Proof. The center of mass of the impulse response of a symmetric filter $f_s(x, y)$ is mean zero, from the definition of a symmetric filter.

$$E_{f_s}\{[x, y]\} = \frac{\sum_{x,y}[x, y] f_s(x, y)}{\sum_{x,y} f_s(x, y)} = [0, 0]$$

Thus by linear superposition, there is no net displacement of the centre of mass $E_f[x, y]$ □

Theorem 2.3 (Antisymmetric convolution generally displaces centre of mass.)**.** *The expected displacement* $E[dx, dy]$ *of the center of mass of a rectified activation image* $f(x, y) \in [0, +\infty]$ *following convolution with an antisymmetric filter* $f_a(x, y)$ *and rectification is generally non-zero*

$$E[dx, dy] = \frac{\sum_{x,y}[x, y] f(x, y)}{\sum_{x,y} f(x, y)} - \frac{\sum_{x,y}[x, y] ReLU\left[f(x, y) * f_a(x, y)\right]}{\sum_{x,y} ReLU\left[f(x, y) * f_a(x, y)\right]}$$

Proof. The centre of mass of an antisymmetric filter f_a impulse response following rectification $ReLU[f_a]$ equals the centre of mass of the positively signed weights $f_a^+ = \max\{f_a, 0\}$

$$\{[dx, dy]\} = E_{ReLU[f_a]}\{[x, y]\} = \frac{\sum_{x,y}[x, y] ReLU[f_a(x, y)]}{\sum_{x,y} ReLU[f_a(x, y)]} = \frac{\sum_{x,y}[x, y] f_a^+(x, y)}{\sum_{x,y} f_a^+(x, y)}$$

Thus by linear superposition, rectified convolution with an antisymmetric filter will generally lead to a displacement $[dx, dy]$ in the centre of mass of an activation image. □

The average centre of mass displacement due to antisymmetric filtering relates to the spatial arrangement of positively signed coefficients $f_a^+ = \max\{f_a, 0\}$. The maximum displacement for a kernel of size $(Width, Width)$ pixels is $dx_{max} = (Width - 1)/2$ which occurs when all positively signed weights lie along a single outer edge of the filter. The minimum displacement of $dx_{min} = 0$ occurs when they are arranged symmetrically about the origin, e.g. a saddle-point such as the DCT coefficient $D_{1,1}$.

2.2.4 2D Filter Structure, $\Sigma + \nabla$ Approximation and Visualization

The previous section presented general notions of symmetric and antisymmetric filters in 2D, which may generally take on a variety of unique patterns, *e.g.* DC Σ, gradients ∇ and higher order patterns for larger filter sizes. Here we show how these may be generally represented as components of the discrete cosine transform (DCT) frequency transform, as is commonly done in image and video compression, which lend themselves to an intuitive interpretation. To our knowledge, this is the first work investigating the symmetry and antisymmetry of the DCT spectrum.

In general, an $N \times N$-pixel filter may be represented as a sum of N^2 discrete cosine transform (DCT) coefficients, each of which may be purely symmetric (S), purely antisymmetric (a) or mixed symmetric + antisymmetric (M). Figure 2.6 a) shows the DCT basis functions up to index or wave number $(u, v) = (4, 4)$. Figure 2.6 b) shows the pattern of symmetry ascribed each DCT basis. We may observe that where one or two wave numbers (u, v) are odd, the DCT basis is antisymmetric (a). If both indices (u, v) are even and equal $u = v$, *i.e.* along the main diagonal, the basis is symmetric (S). If both indices (u, v) are even but unequal $u \neq v$, then the basis is mixed symmetric and antisymmetric (M).

Figure 2.7 shows the symmetric and antisymmetric components of DC bases. Note for symmetric components in Figure 2.7 a), off-diagonal coefficient pairs (u, v) and (v, u) are equal. Note that for antisymmetric components in Figure 2.7 b), off-diagonal coefficient pairs (u, v) and (v, u) differ by an in-plane rotation angle of either $\pi/2$ or $3\pi/2$ and may be used as quadrature pairs to generate intermediate rotations.

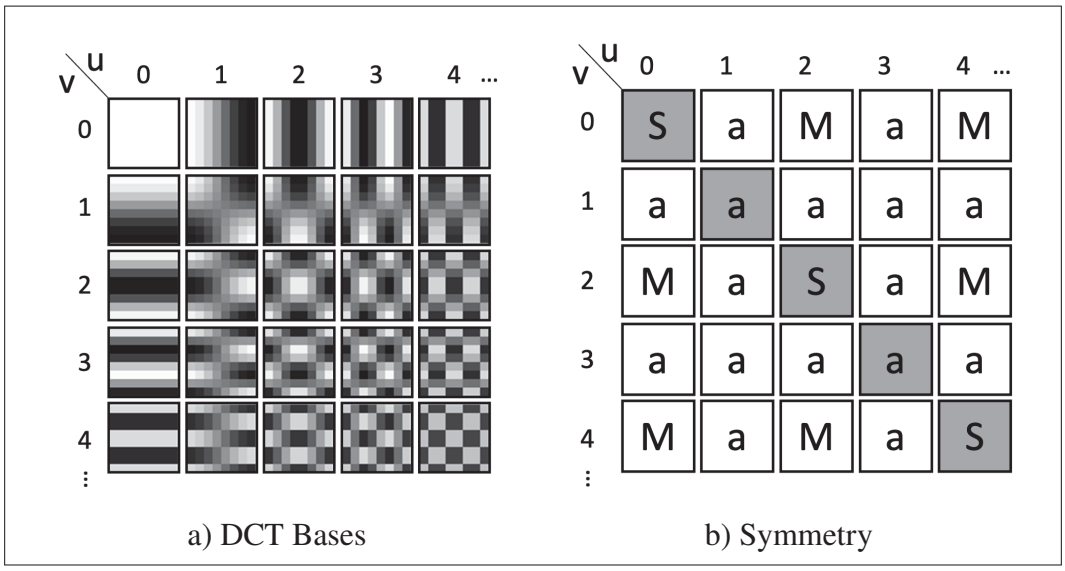


Figure 2.6 a) shows the DCT bases up to index (4,4), b) indicates whether the basis is symmetric S, antisymmetric a or mixed M

In practice, images or kernels representing natural images are dominated by low frequency DCT components. Particularly in our framework for small filters, *i.e.* 3×3 pixels, the symmetric

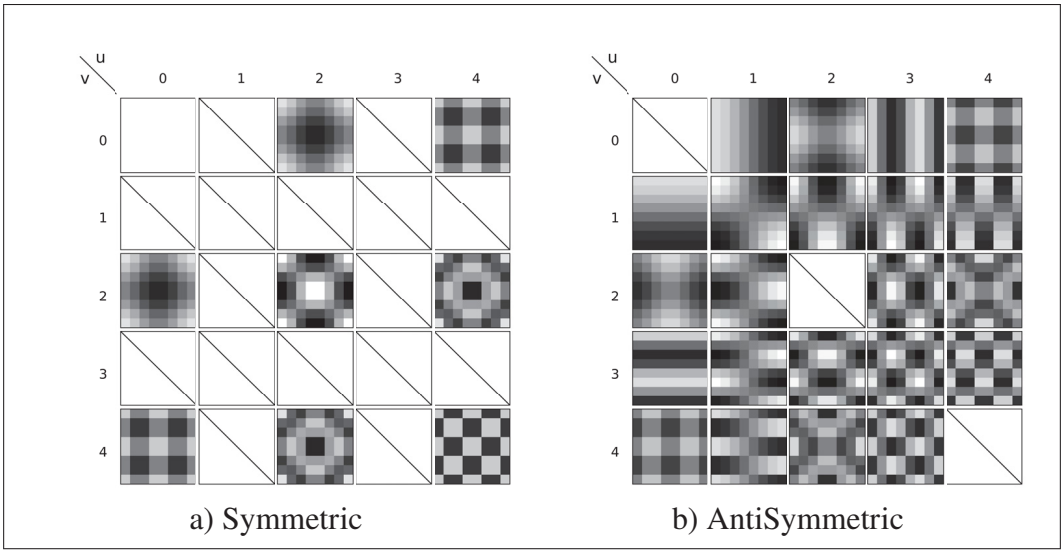


Figure 2.7 Illustrating the symmetric a) and antisymmetric b) components of DCT bases, black indicates the component is zero

component may be approximated by DC or sum:

$$f_s(x,y) = \sum_{u,v \in Even} \omega_{u,v} D_{u,v} \approx \omega_{0,0} D_{0,0} = \omega_{0,0} \Sigma. \tag{2.12}$$

While the antisymmetric component may be approximated by the gradient or difference ∇, where the angular parameter θ defines the gradient orientation as a linear mix between horizontal ∇_x and vertical ∇_y gradient components:

$$f_a(x,y) = \sum_{u,v \in Odd \cup u \neq v} \omega_{u,v} D_{u,v}, \tag{2.13}$$

$$\approx \omega_{0,1} D_{0,1} + \omega_{1,0} D_{1,0} = \cos\theta \nabla_x + \sin\theta \nabla_y. \tag{2.14}$$

This approximation is validated in experiments, where retraining CNNs with only three of nine filter components $(\Sigma, \nabla_x, \nabla_y)$ results in greater than 90% of baseline accuracy for typical networks, e.g. VGG and ResNet.

Furthermore, CNN kernels may be visualized in a 3D space spanned by the signed magnitudes of $(\Sigma, \nabla_x, \nabla_y)$ components as shown in Figure 2.8, where the DC component Σ defines the vertical symmetric axis, and the gradient components (∇_x, ∇_y) define a horizontal plane.

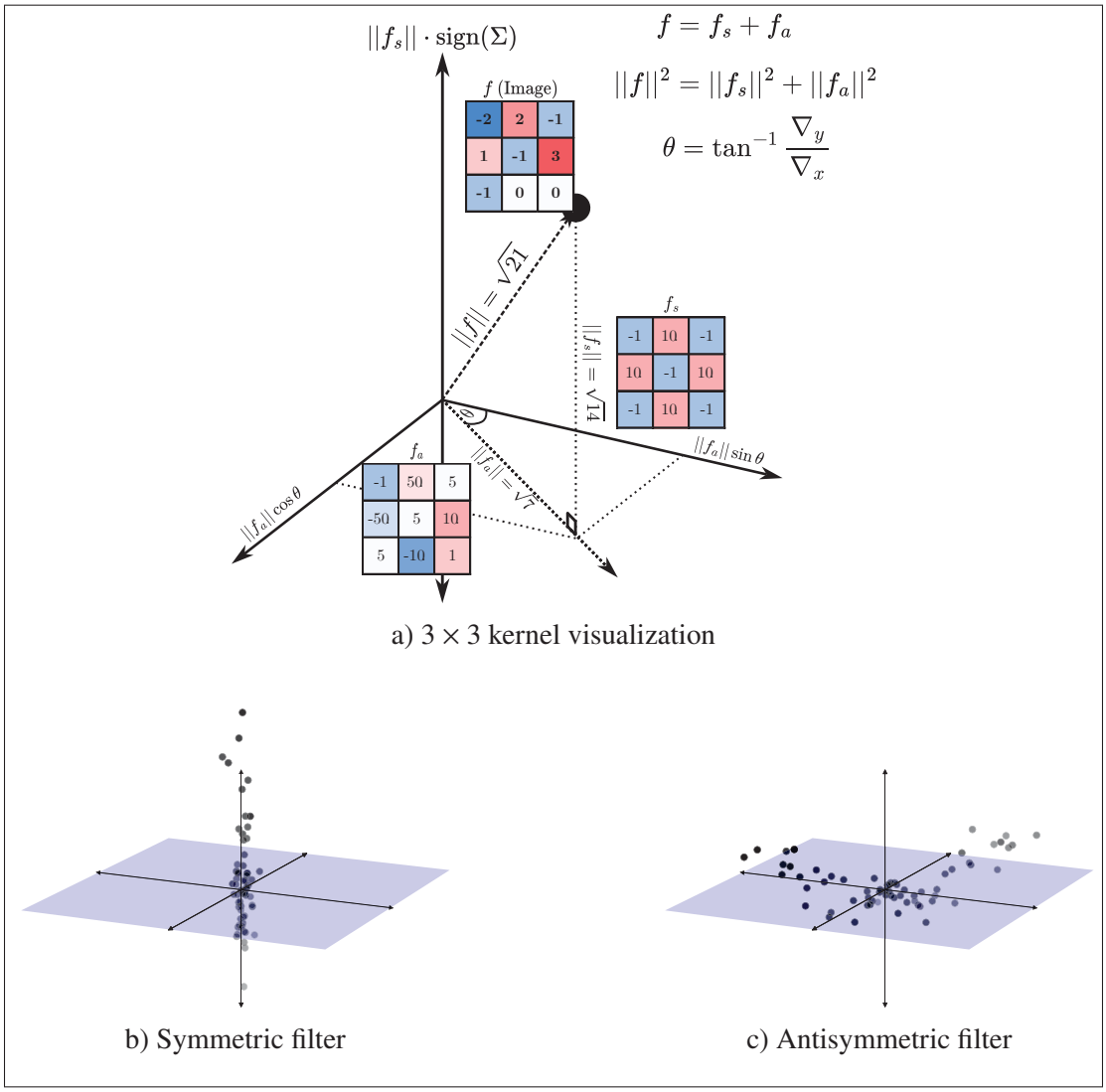


Figure 2.8 a) Visualization of a single-channel 3×3 kernel f in 2D image space with squared magnitude $||f||^2 = 21$ decomposed into a symmetric axis (vertical) of squared magnitude $||f_s||^2 = 14$ and an antisymmetric plane (horizontal) of squared magnitude and $||f_a||^2 = 7$. b) and c) show examples of dominantly symmetric and antisymmetric filters in a deep CNN layer

2.3 Demonstrating Information Propagation under Rectified Convolution

The effects of linear convolution with various kernels are well-known. Symmetric filters produce isotropic responses, treating image content equally regardless of orientation, whether for blurring (*e.g.* Gaussian, averaging) or edge detection (*e.g.* Laplacian). In contrast, antisymmetric filters, such as gradient-based filters (*e.g.* Sobel), respond maximally to edges aligned with their orientation and minimally to orthogonal edges.

As mentioned in Chapter 1, studies of propagation resulting from rectified convolution have been limited to Effective Receptive Field size (Luo *et al.*, 2016), and have not considered the various modes by which information propagates, for example, displacement as a result of gradient filtering. This section aims to understand these properties. Here, we demonstrate for the first time that, under rectified convolution, the maximum speed at which the centre of mass of activation information displaces is a monotonic function of the ratio β^2 of antisymmetric to total filter energy, and for 3×3 filters is almost a linear function of β^2.

In the first set of experiments we demonstrate the effect of "unipolar kernels", by repeatedly convolving a test pattern with a kernel with either a fixed or left-right alternating gradient orientation. As we observe in Figure 2.9a, a purely symmetric kernel ($\beta^2 = 0$) produces a diffusion effect, where the content of the filtered image is radiated isotropically. When an antisymmetric gradient component ∇ of fixed orientation is mixed with symmetric DC Σ in equal proportions ($\beta^2 = 0.25$), the circle translates rightwards, with a noticeable foreshortening along the axis of translation in Figure 2.9b ($\sigma_0 = 11$, $\sigma_{100} = 10.83$). With a purely antisymmetric component ($\beta^2 = 1$) as in Figure 2.9c, a leading edge of approximately single pixel thickness propagates to a maximum distance of $100 \times \frac{Width}{2}$, and the bulk of the circle disappears.

Figure 2.10 shows the result of repeated convolution with an antisymmetric gradient component $\pm\nabla$ of alternating orientation, which is similar, however, without translation. With a filter mixing equal symmetric DC Σ and alternating gradient $\pm\nabla$ components ($\beta^2 = 0.25$) as in Figure 2.10a, the circle remains in place with a noticeable foreshortening along the axis of translation ($\sigma_0 = 11$, $\sigma_{100} = 10.74$). With a purely antisymmetric component ($\beta^2 = 1$) as in Figure 2.10b, the right

edge of the circle vibrates in place at a width of approximately single pixel thickness, and the bulk of the circle disappears.

Propagation resulting from convolution higher order filters may be interesting to study in future works, however, the three modes of propagation which can follow from convolving with Σ and ∇ allow for a natural interpretation of information.

During propagation experiments, the test pattern translates or propagates in the image plane according to the mixing ratio $\beta^2 = \frac{\|f_a\|^2}{\|f\|^2}$ of the squared antisymmetric energy $\|f_a\|^2$ vs the total squared filter energy $\|f\|^2$. Specifically, the test pattern centre of mass remains stationary for symmetric filters with $\beta^2 = 0$, propagates at a maximum speed of $c = \frac{Width-1}{2}$ for purely antisymmetric filters $\beta^2 = 1$, and at intermediate speeds for mixed filters $0 < \beta^2 < 1$.

Figure 2.11b plots the squared distance $(\frac{dx}{dx_{max}})^2$ travelled by the image test pattern vs the mixing ratio $\beta^2 = \frac{\|f_a\|^2}{\|f\|^2}$, for several filter sizes. The distance dx is measured according to the test pattern centre of mass following $t = 10$ iterations, and is normalized to a range of $[0, 1]$ by the maximum possible distance $dx_{max} = \frac{Width-1}{2}$ set according to the filter $Width$ in pixels for three different filter sizes. We see that maximal propagation distance is achieved under ReLU, which enables directional flow of positive activations.

See Chapter I for additional propagation demonstrations.

2.3.1 Demonstrating Attenuation due to Orientation in Sequential Convolutions

Another property of note we investigate is the effect of sequential convolution (with ReLU) of antisymmetric gradient kernels. The convolution operator $*$ is symmetric and associative, meaning that in a chain of convolutions, kernels can be pre-computed in any order, rather than in sequence, as demonstrated in Equation (2.15) and Equation (2.16).

$$I * f_1 = f_1 * I \tag{2.15}$$

$$f_2 * (I * f_1) = I * (f_1 * f_2). \tag{2.16}$$

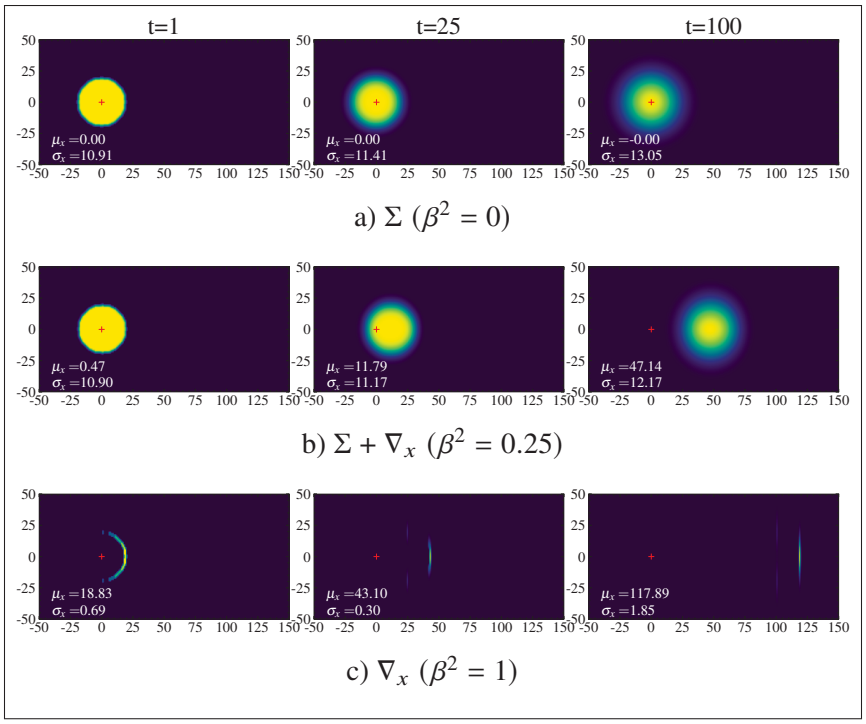


Figure 2.9 Demonstrating the effect of repeated convolution+ReLU of a circle ($r = 19$) test pattern over different types of 3×3 kernels (DC and Gradient)

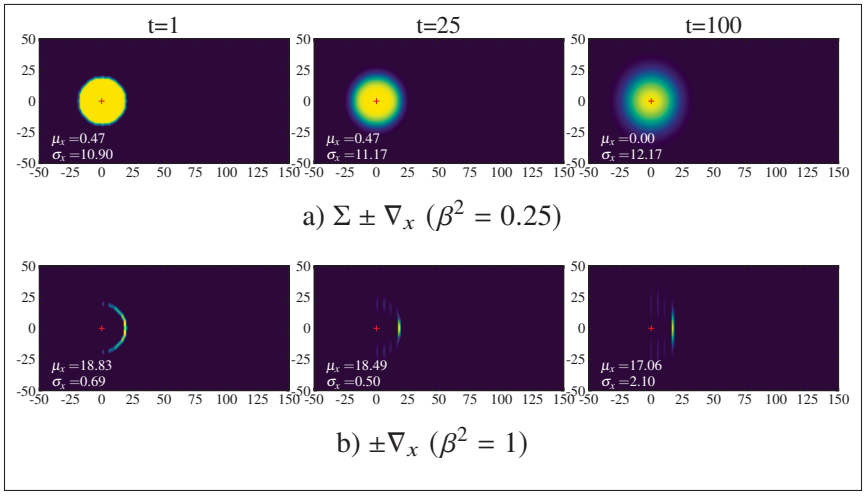


Figure 2.10 Demonstrating the effect of repeated convolution with ReLU with alternating orientation

In a typical CNN, however, the associative property does not hold true due to the ReLU layer, which breaks the linearity. Nonetheless, the expected magnitude of the result of sequential

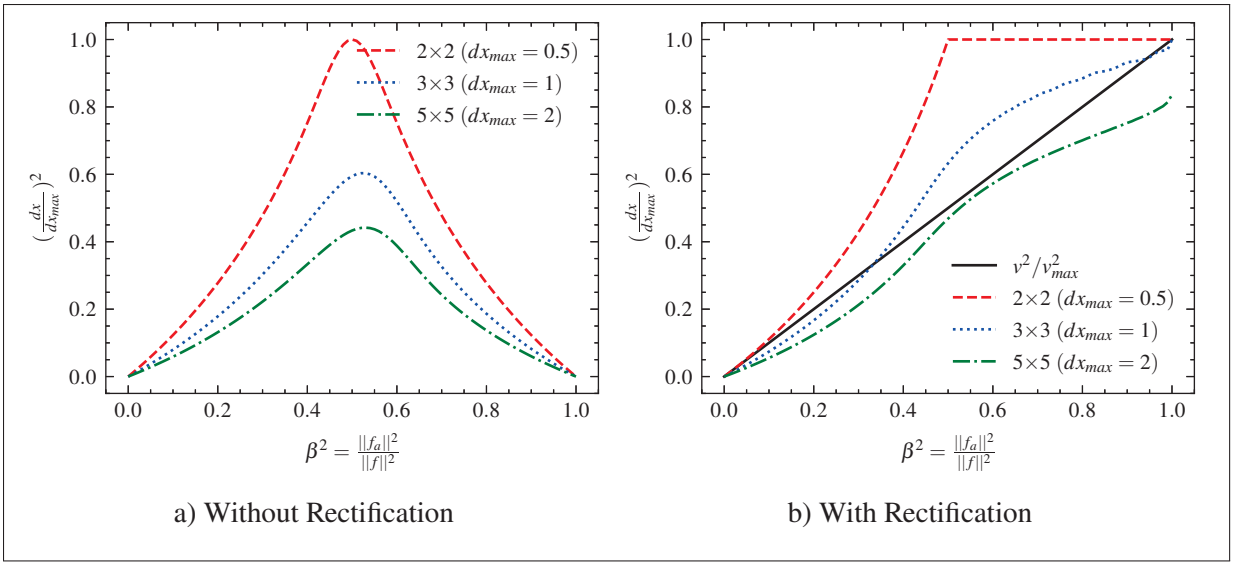


Figure 2.11 Distance of Propagated information (100 sequential convolutions), relative to maximal propagation distance ($dx_{max} = \frac{Width-1}{2}$), for various β^2. Results showed for a) without any non-linearities (ReLU) and b) with ReLU

convolution followed by ReLU may be expressed in terms of the in-plane filter orientation θ as follows. Let θ_I represent the angle of the local image content and θ_i represent the orientation of a gradient filter ∇_i. The magnitude $\|I_1\|$ of the dot product between the image I and filter ∇_i, which is proportional to the cosine of the angular difference $(\theta_I - \theta_i)$

$$||ReLU(I \cdot \nabla_i)|| \propto \cos(\theta_I - \theta_i), \tag{2.17}$$

and thus information is attenuated due to the angular difference. For N filters in a sequence, the attenuation is proportional to the product of angular difference cosines:

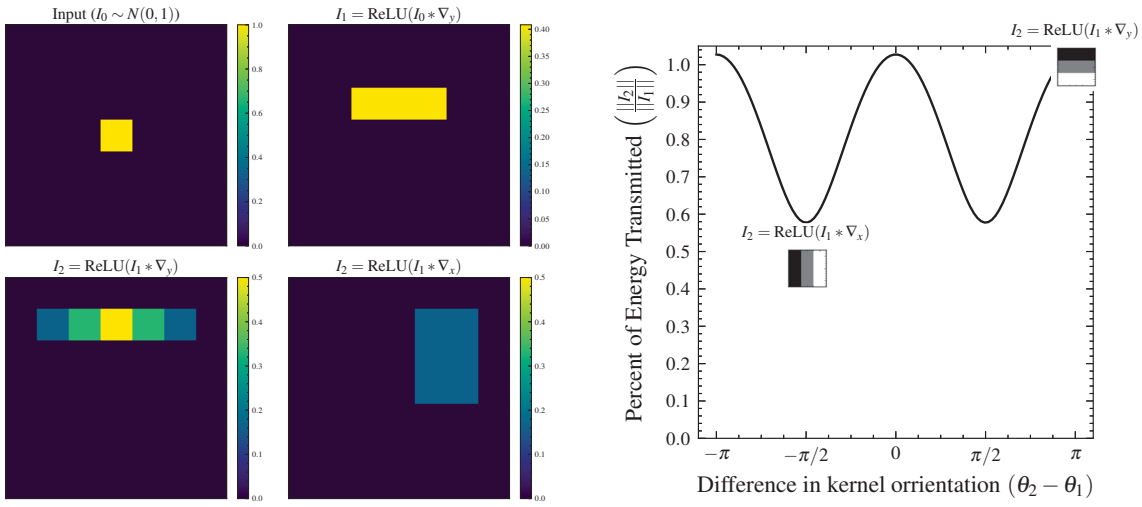
$$\|I_N\| \propto \prod_{i}^{N} \cos(\theta_I - \theta_i). \tag{2.18}$$

In Equation (2.18), the product of cosines diminishes rapidly in the case of significant angular differences, and thus for oriented image information to pass through a sequence of odd CNN filter components, the angular difference must be small both between filters and the image

$\theta_i \approx \theta_I$ and between different filters $\theta_i \approx \theta_j, i \neq j$ in a sequence. We note that activation image magnitude is attenuated following sequential convolution of two gradient filters ∇_1 and ∇_2 with an angular difference $\theta_1 - \theta_2$ varied over a range of $[-\pi, \pi]$.

We demonstrate in Figure 2.12a the experiment setup. Figure 2.12b demonstrates that, independently of the input image, sequential antisymmetric filtering will result in maximum signal attenuation for filters at right angles $\theta_2 = \theta_1 \pm \pi/2$, and minimum attenuation for translation at identical angles $\theta_2 = \theta_1$.

We repeat this experiment in Sections I.2.3 and I.2.4 for higher order antisymetric kernels $D_{1,2}$ and saddle kernels $D_{1,1}$. We again observe that attenuation follows a sinusoidal curve, where subsequent convolution by orthogonal kernels has maximal attenuation. This demonstrates that angular attenuation generally applies to off-diagonal antisymmetric pairs of the DCT, which differ by $\theta = \frac{\pi}{2}$.



a) Demonstration b)

Figure 2.12 Demonstration of sequential convolution of orthogonal, identical filtering (a). Activation magnitude following a sequence of two gradient filters ∇_1 and ∇_2 with varying angular difference $\theta_1 - \theta_2$ on a circular test pattern (b). Filter ∇_2 is steered according to the steering expression : $\nabla_2 = \nabla_x \cos(\theta_2) + \nabla_y \sin(\theta_2)$

2.4 Randomly Initialized Filters

Convolutional neural networks are typically initialized with random weights drawn from simple, zero-mean distributions (*e.g.* Gaussian or uniform), chosen to preserve signal variance across layers before training, as in Glorot initialization (Glorot & Bengio, 2010) and He initialization (He *et al.*, 2015). For this reason, we analyze in this section the symmetric and antisymmetric energy decomposition of randomly initialized convolutional kernels.

This allows us to characterize the distribution of β^2, the ratio of antisymmetric to total energy, under random weights, thereby establishing a baseline against which trained networks can be compared.

We characterize the energy of full kernels and their symmetric and antisymmetric components in terms of χ^2 distributions. Thus leading us to describe β^2 as a Beta distribution, parameterized by the size of the kernel. We find that as kernel sizes increase, the β^2 distribution at initialization skews towards being predominantly antisymmetric.

Given an $N \times N$ image I with each pixel randomly sampled from a zero-mean Normal distribution $I_i \sim N(0, \sigma_I^2)$, we find that its total energy $||I||^2$ follows a scaled-χ^2_N distribution.

$$||I||^2 = \sum_{i=1}^{N^2} I_i^2 \sim \sigma_I^2 \chi^2_{N^2} \tag{2.19}$$

As described in Definition 2.1, we decompose I into its symmetric component I_s by averaging the k pixels in each set of equidistant points from the center. A set can either contain 1 pixel (center), 4 pixels (corner or center of the edges) or 8 pixels (other). Let C_r be a set of all pixel coordinates that are at a radius r from the center $(0, 0)$ of I.

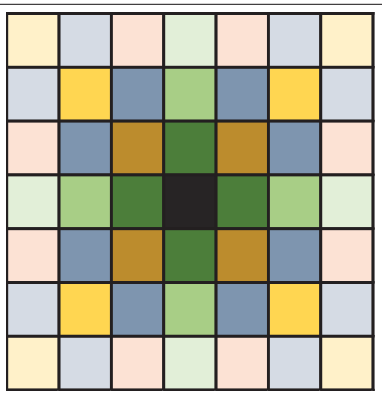
$$C_r = \left\{ (x, y) \in \mathbb{Z}^2 \middle| \sqrt{x^2 + y^2} = r \right\} \tag{2.20}$$

Let C be the set of sets of points equidistant from the center of I. In other words, let C be the set containing all C_r.

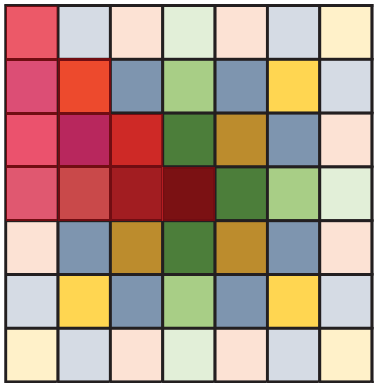
$$C = \{C_r | r \in \mathbb{R}_{\geq 0}\} \tag{2.21}$$

For a symmetric component I_s , let $I_s^{(r)}$ be a given pixel intensity which was calculated by averaging I over the coordinates in C_r.

$$I_s^{(r)} = \frac{1}{|C_r|} \sum_{(x,y) \in C_r} I(x, y) \tag{2.22}$$



a) $N \times N$ symmetric image. Each distinct radius is denoted by a unique colour.



b) Upper half-quadrant (in red) for counting unique radii in the symmetric image.

Figure 2.13 An example $N \times N$ image ($N = 7$) demonstrating how to count unique radii

Let R be the set of radii for which the sets C_r are defined. We can define the energy, $||I_s||^2$ of I_s as :

$$||I_s||^2 = \sum_{r \in R} |C_r| \left(I_s^{(r)}\right)^2 \tag{2.23}$$

Since each pixel of the initial image I follows a Normal distribution $N(0, \sigma_I^2)$, following Equation (2.22), we model $I_s^{(r)}$ as

$$\begin{aligned} I_s^{(r)} &\sim \frac{1}{|C_r|} N(0, |C_r|\, \sigma_I^2) \\ &\sim \frac{1}{\sqrt{|C_r|}} N(0, \sigma_I^2) \end{aligned} \tag{2.24}$$

$||I_s||$ follows a χ distribution, with d.o.f. $|R|$. We find $|R|$ to be the total number of pixels in one half-quadrant of the image, as shown in red in Figure 2.13b. We can compute $|R|$ with:

$$|R| = \frac{\left(\lceil \frac{N}{2} \rceil + 1\right) \lceil \frac{N}{2} \rceil}{2}, \tag{2.25}$$

where in Equation (2.25) $\lceil \cdot \rceil$ is the ceiling operator. Following the symmetric energy definition in Equation (2.23), we model $||I_s||$ as:

$$\begin{aligned} ||I_s||^2 &\sim \sigma_I^2 \chi^2_{|R|} \\ ||I_s||^2 &\sim \sigma_I^2 \, \chi^2_{\frac{\left(\lceil \frac{N}{2} \rceil + 1\right) \lceil \frac{N}{2} \rceil}{2}} \end{aligned} \tag{2.26}$$

$||I_a||$ also follows a χ distribution. We know from Definition 2.3 that $||I||^2 = ||I_a||^2 + ||I_s||^2$, therefore,

$$||I_a||^2 = ||I||^2 - ||I_s||^2 \tag{2.27}$$

$$||I_a||^2 \sim \sigma_I^2 \chi^2_{N^2} - \sigma_I^2 \, \chi^2_{|R|} \tag{2.28}$$

$$\sim \sigma_I^2 \, \chi^2_{N^2 - |R|} \tag{2.29}$$

With a description of $||I_a||$ and $||I_s||$, defined as following χ distributions, *e.g.* following random initialization, we can describe β^2 (see Equation (2.8)) as statistically following a Beta distribution.

$$\beta^2 = \frac{||I_a||^2}{||I_a||^2 + ||I_s||^2} \tag{2.30}$$

$$\beta^2 \sim \frac{\chi^2_{N^2-|R|}}{\chi^2_{N^2-|R|} + \chi^2_{|R|}} \tag{2.31}$$

$$\sim \text{Beta}\left(\frac{N^2 - |R|}{2}, \frac{|R|}{2}\right) \tag{2.32}$$

See Figure 2.14 for examples of the Beta distribution for different kernel sizes. We find that as kernel sizes increase, their β^2 distribution at initialization skews towards being predominantly antisymmetric.

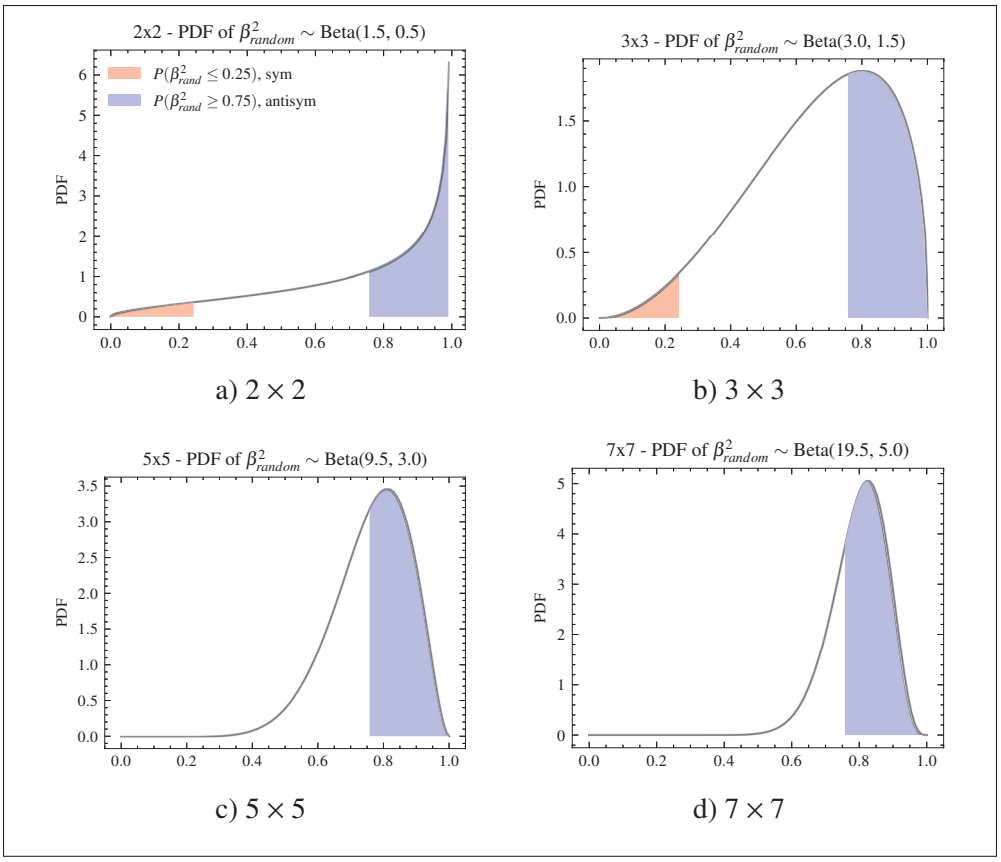


Figure 2.14 Probability distribution (PDF) of β^2 of a randomly initialized $N \times N$ convolutional kernel. We highlight the areas under the curve, designating the (anti)symmetric regions, to show that a randomly initialized kernel starts predominantly antisymmetric

CHAPTER 3

EXPERIMENTS

This chapter empirically validates the main hypothesis of this thesis: that the dominant propagation effects induced by rectified convolution are governed primarily by the lowest-order symmetric and antisymmetric components of 3×3 kernels, namely the DC component Σ and the first-order gradients (∇_x, ∇_y). In Chapter 2, we demonstrated how these components correspond to different modes of propagation in image space: Σ drives diffusive (smoothing) behaviour, while (∇_x, ∇_y) induces oriented translation. The experiments in this chapter, therefore, ask a concrete question: do trained CNNs in fact learn filters whose energy concentrates in this low-frequency subspace, such that $\Sigma + \nabla$ is sufficient to capture most of their functional behaviour?

We begin by confirming that kernel energy is concentrated in the lowest-frequency components and that near-optimal accuracy is achievable with only low-frequency subsets. Beyond accuracy, we also track how symmetry and antisymmetry evolve with depth: early-layer filters are predominantly antisymmetric, while deeper layers exhibit increasingly symmetric behaviour. Finally, we examine the geometric organization of antisymmetric components and find that filters develop a correlated bipolar structure in orientation, meaning that channels within a filter tend to share a common or opposite dominant orientation. This orientation relationship also extends across depth: kernels preferentially activate in response to similarly oriented filters from preceding layers, while suppressing activations from orthogonally oriented filters.

First, in Section 3.1, we study training and evaluation under spectral constraints. We introduce a DCT-parameterized convolutional layer in which each kernel is generated as a linear combination of N DCT bases with learnable coefficients. This allows us to progressively restrict frequency bands while keeping the rest of the architecture unchanged, thereby isolating the contribution of specific components to classification performance. We evaluate (i) the spectral energy distribution of ImageNet-trained kernels in VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016), (ii) training from scratch on CIFAR-100 with varying numbers of preserved

DCT coefficients, and (iii) fine-tuning ImageNet models after truncating their learned kernels to low-frequency subsets. Across these experiments, we find that a small number of low-frequency components recovers most of the baseline accuracy, with $(\Sigma, \nabla_x, \nabla_y)$ providing a strong approximation.

Second, in Section 3.2, we turn from performance to structure. Using the symmetric/antisymmetric decomposition framework, we characterize how kernels organize across layers in trained VGG16 and ResNet50 models. We show that training drives many kernels toward being predominantly symmetric or antisymmetric, and that antisymmetric kernels exhibit orientation correlation. Furthermore, this orientation relationship between filter-channels extends across depth; kernels preferentially activate in response to similarly oriented filters from preceding layers, while suppressing activations from orthogonally oriented filters.

Together, these experiments support the view that low-order symmetric and antisymmetric components are not only a convenient basis for mechanistic interpretation, but also a compact representation that largely preserves the functional capacity of learned CNN filters.

3.1 Training Experiments

The main theme of this thesis is interpreting kernel weights in terms of symmetric and antisymmetric components. These may be represented in terms of the DCT frequency decomposition, and approximated by small sets of the lowest low-frequency components, which are dominant in natural images (Ruderman & Bialek, 1993) and emphasized in image compression. Here, we evaluate CNN training with various subsets of DCT components to assess their impact on classification accuracy. We hypothesize that the majority of accuracy may be achieved by small subsets of low frequency components. We find that a 3-component representation as previously described in Chapter 2 including the symmetric DC component Σ the antisymmetric first-order oriented gradients (∇_x, ∇_y) leads to an effective approximation capturing $> 90\%$ of classification accuracy.

In Chapter 2, we proposed an approximation for 3×3 CNN kernel interpretation, based on three parameters: symmetric energy, antisymmetric energy and gradient orientation θ. Naturally, reducing from 9 parameters to 3 would seem to cause a significant loss of information, so in this section, we validate the hypothesis that the bottom 3 DCT components (Σ, ∇_x, ∇_y) approximate the majority of the CNNs representation.

The following experiments evaluate how individual frequency components contribute to network behaviour. The setup is illustrated in Figure 3.1. In a standard convolutional layer, each input channel C_i is convolved with a distinct $k \times k$ kernel f_i, and the resulting feature maps are summed to produce the filter output (see Equation (3.1)).

Because any kernel or image can be represented as a linear combination of DCT bases, we introduce a modified convolutional layer that incorporates this decomposition. During forward propagation, each kernel is generated as a weighted sum of N DCT bases, where the weights ω are learned parameters (see Equation (3.2)).

$$Y = \sum_{j=0}^{C} f_j \ * \ C_j \tag{3.1}$$

$$= \sum_{j=1}^{C} \left(\sum_{i=1}^{N} \omega_i \, DCT_i \right) \ * \ C_j \tag{3.2}$$

When using the full spectrum ($N = k^2$), the layer behaves identically to a standard convolution. Since kernel generation (eq. (3.2)) is a linear operation, backpropagation remains unchanged, meaning any convolutional layer in a trained CNN can be replaced by its DCT-based counterpart without altering the model's behaviour. By varying N, we analyze how each frequency band contributes to network performance.

We consistently find that most of the representation in trained CNNs is encoded by the lowest three frequency components, corresponding to the DC (Σ) and first-order gradient bases (∇_x, ∇_y).

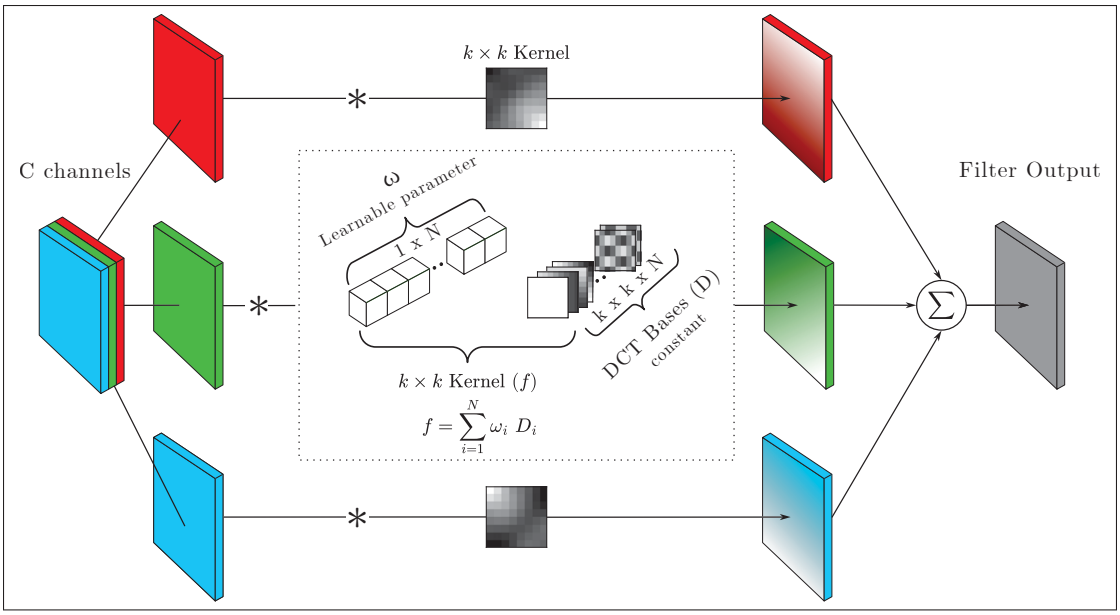


Figure 3.1 Illustrating a DCT convolution layer, where each kernel is a linear sum of N DCT basis components scaled by a learnable weighting parameter ω. With the exception of the kernel generation step, the overall mechanism is identical to a standard convolution layer

3.1.1 Energy Distribution of ImageNet-trained Kernels

Figure 3.2 a) and b) show the DCT spectral distributions for ImageNet-trained VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016) networks averaged over all layers. As we can clearly see from Figure 3.2, the majority of the energy is shared between DC Σ (symmetric) and gradient ∇ (antisymmetric) after training, whereas distributions following random initialization are uniform (Figure 3.2c). Chapter II provides spectral energy distributions for all individual 3×3 convolution layers for ImageNet-trained VGG16 (Figure II-1) and ResNet50 (Figure II-2), showing similar distributions throughout the networks.

3.1.2 Training from Scratch - Cifar100 Dataset

This experiment consists of training a VGG16 (Simonyan & Zisserman, 2015) and Resent20 He *et al.* (2016) models with various numbers of DCT coefficients, from a single DC (Σ) parameter to 9 total components (full spectrum). The parameters used are DCT kernel weights $\boldsymbol{\omega}$, which

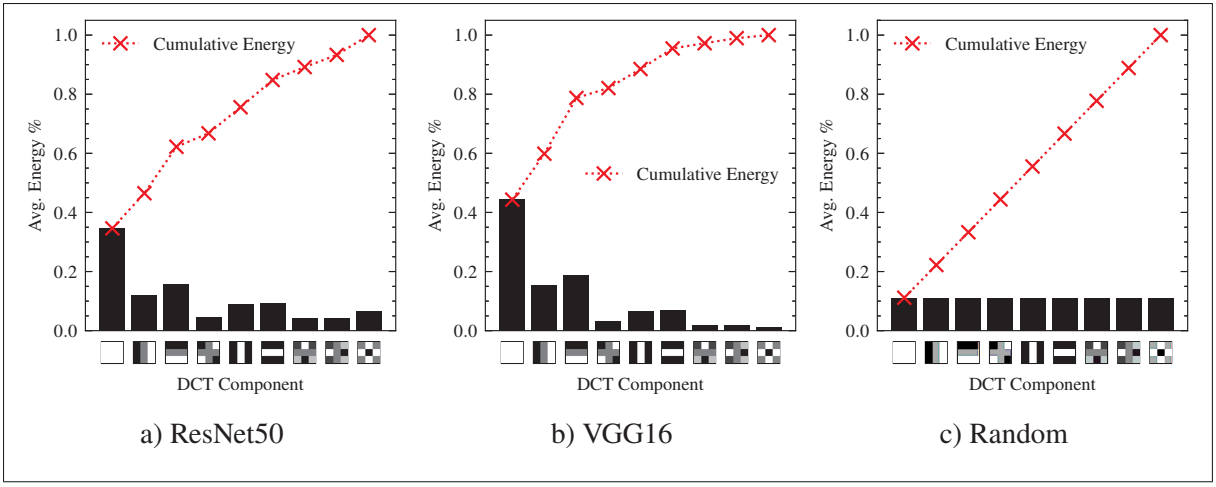


Figure 3.2 Spectral DCT decomposition ω_i of all 3×3 convolutional filters in all layers of a) ResNet50 and b) VGG16. We find that in both models the majority of the weights are comprised of low order DC and Gradients ($\Sigma + \nabla$)

are updated during backpropagation and inverse transformed into filters for forward propagation (see Figure 3.1). For all 6 runs, we use the same hyperparameters which yielded the best accuracy for the baseline. As we can see in Table 3.1, the run with only DC (Σ) and low-order gradient (∇_x, ∇_y) components, VGG16 achieves 66.64% validation accuracy. Introducing the mixed component $D_{1,1}$ results in a slightly higher accuracy than without (68.23%), however, the greatest increase comes when using the $D_{0,2}$ and $D_{2,0}$ kernels as well (71.62%). We observe the same behaviour when training ResNet20 (see Table 3.2).

Table 3.1 Training VGG16 on CIFAR-100 (Krizhevsky *et al.*, 2009) using convolutional kernels composed of progressively additional high-order DCT components. We find that only 3 low-frequency components (underlined) contribute to 91% of VGG16 baseline accuracy.

Number of DCT Components	Val-Accuracy (± std)	% of Baseline
1 (Σ)	0.3247 ± 0.0052	0.44
3 (Σ, ∇_x, ∇_y)	0.6664 ± 0.0017	0.91
4	0.6823 ± 0.0039	0.93
6	0.7162 ± 0.0055	0.98
8	0.7294 ± 0.0032	0.99
9 (Baseline)	**0.7299 ± 0.0019**	**1.00**

Table 3.2 Training Resnet20 on CIFAR-100 (Krizhevsky *et al.*, 2009) using convolutional kernels composed of progressively additional high-order DCT components. We find that only 3 low-frequency components (underlined) contribute to 92% of Resnet20 baseline accuracy.

Number of DCT Components	Val-Accuracy (± std)	% of Baseline
1 (Σ)	0.4301 ± 0.022	0.63
3 ($\Sigma, \nabla_x, \nabla_y$)	0.6277 ± 0.0025	0.92
4	0.6413 ± 0.0084	0.94
6	0.6675 ± 0.0074	0.98
8	0.6759 ± 0.0069	0.99
9 (Baseline)	**0.6805 ± 0.0104**	**1.00**

To test the boundary scenario, we also examine the inverse case for completeness (see Table 3.3). Instead of training a CNN with fewer high order DCT coefficients, we now try it with fewer low order coefficients. This approach helps rule out the possibility that high-order components alone are responsible for achieving high accuracy.

Table 3.3 Training VGG16 on CIFAR-100 using convolutional kernels composed of progressively fewer low-order DCT components

Number of DCT Components **not** Used	Val-Accuracy (± std)
0 (Baseline, using all components)	**0.7299 ± 0.0019**
1 not using (Σ)	0.657± 0.0045
3 not using ($\Sigma, \nabla_x, \nabla_y$)	0.535 ± 0.0046
4	0.484 ± 0.0055
6	0.306 ± 0.0248
8 (Only using $D_{2,2}$)	0.0273 ± 0.0019

3.1.3 Fine-tuning - ImageNet Dataset

For the remainder of this work, we focus on understanding the mechanisms that emerge through training on the ImageNet (Deng *et al.*, 2009) dataset. In this last experiment, we seek to evaluate the contribution of DCT frequency component subsets toward the classification task on the ImageNet dataset (ILSVRC2012) (Deng *et al.*, 2009). We evaluate VGG16 (Simonyan & Zisserman,

2015) and ResNet50 (He *et al.*, 2016) models (trained on ImageNet) by progressively reducing the number of preserved DCT coefficients representing each 3×3 convolutional kernel, down to a single DC component. First, we project the learned 3×3 kernels onto the DCT basis and retain only the bottom N coefficients (ω). The models are then evaluated using the truncated kernels as shown in Figure 3.1. We observe that with minimal fine-tuning on the preserved coefficients, the models recover performance close to the original baseline accuracy. This indicates that most of the information in learned filters lies in low-order frequencies, and this can perhaps be used in order to compress models or initialize models for efficient learning.

For fine-tuning, both models are trained using SGD for 8 epochs with a learning rate $\eta = 10^{-5}$ and batch size 256. Following the protocol in He *et al.* (2016), during training, we randomly resize the shorter side of each image between $[256, 480]$, then randomly crop a 224×224 patch and apply random horizontal flipping. For validation, we resize the shorter side to 256 and centre crop a 224×224 patch.

As shown in Figure 3.3, both ResNet50 and VGG16 recover performance close to their original baseline accuracies, with a maximal increase in the first epoch. The largest increase is registered for only 3 (Σ , ∇_x and ∇_y) components, where ResNet and VGG16 obtain 92% and 94% of the baseline accuracy, respectively, while those components only account for 62% and 79% of the total spectrum, respectively (Figure 3.2).

These findings suggest that, for interpretability purposes, 3×3 CNN filters can be effectively approximated by DC (Σ) and Gradient (∇), and that higher frequency components have a minor effect. We therefore permit ourselves continue our study using 2 distinct components. The symmetric component, described by a sign and a magnitude, and the antisymmetric component, which consists of a magnitude and an orientation.

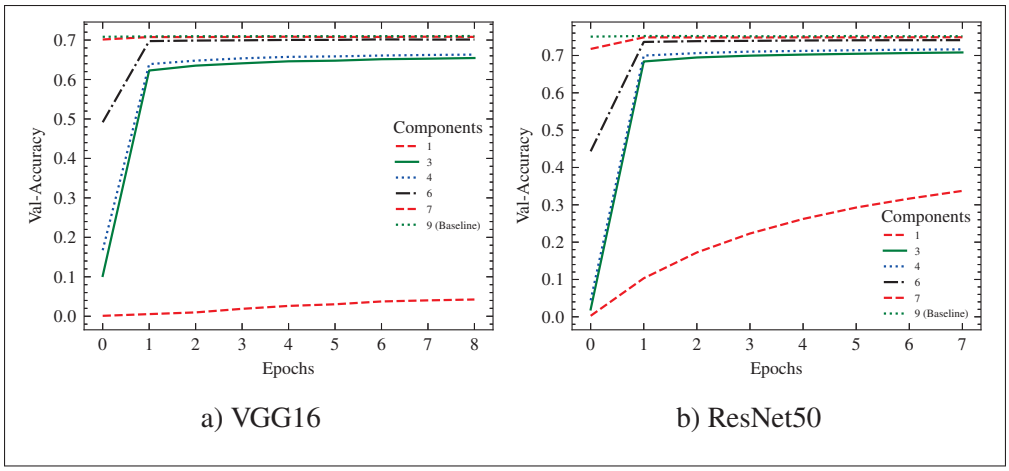


Figure 3.3 Preserving N DCT components of learned 3×3 weights (Trained on ImageNet) and retraining only those components on ImageNet. Note that 3 components ($\Sigma + \nabla$) account for approximately 93% of the baseline representation

3.2 Trained CNN Observations

In this section, we look at the structure of learned convolutional kernels in VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016) (trained on ImageNet (Deng *et al.*, 2009)) through the lens of antisymmetry. We start with an introductory example in Section 3.2.1 where we decompose RGB filters (First layer), in order to apply the framework described in Chapter 2 and find that RGB layers generally emerge to a correlated antisymmetric structure (similarly oriented kernels) or a sparse symmetric structure (coloured filter).

We find that throughout the network, filters develop a correlated bipolar structure along antisymmetric orientations, meaning that the channels within a filter tend to share a common or opposite dominant orientation. Furthermore, this orientation relationship between filter-channels extends across depth; kernels preferentially activate in response to similarly oriented filters from preceding layers, while suppressing activations from orthogonally oriented filters.

We limit our analysis to the top-10% largest filter per layer ranked by ℓ^2-norm. This is due to the assumption that a majority of trained filters are actually inconsequential to a neural network's

performance and that the ℓ^2-norm of a filter is a satisfactory metric for determining its relevance (Li *et al.*, 2017).

Our observations were made on pretrained VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016) trained on ImageNet (Deng *et al.*, 2009) and weights provided by Keras (Chollet *et al.*, 2015).

3.2.1 Randomly Initialized versus Trained Kernels

As an introductory analysis, we can look at the RGB layers in VGG16 and ResNet50; at random initialization and at the end of training on ImageNet (Deng *et al.*, 2009). As described in Chapter 2, we decompose each kernel of a filter into symmetric and antisymmetric components, and plot their energies as demonstrated in Figure 2.8. In Figures 3.4a and 3.5a we see that at random initialization, as expected, filters do not have a specific structure in either the symmetric or antisymmetric components. After training, we notice that filters either develop a strong orientation component (antisymmetric) or become strongly symmetric. We observe that antisymmetric RGB filters tend to be unipolar (correlated in orientation), and symmetric filters tend to be sparse and coloured.

Visualizing individual trained RGB filters (Figures 3.4 and 3.5) leads us to the well known conclusion: input layer filters are primarily comprised of low-frequency Gabor-like blob/colour and edge detectors (Chowers & Weiss, 2023; Yosinski *et al.*, 2014, 2015; Zeiler & Fergus, 2014). A closer look at the antisymmetric projection (orientation component) of certain filters (Figure 3.8) reveals that these RGB filters actually emerge to a correlated antisymmetric structure (similarly oriented kernels) which correspond to grayscale filters, or anti-correlated channels (sparsely oriented kernels) exhibiting colour.

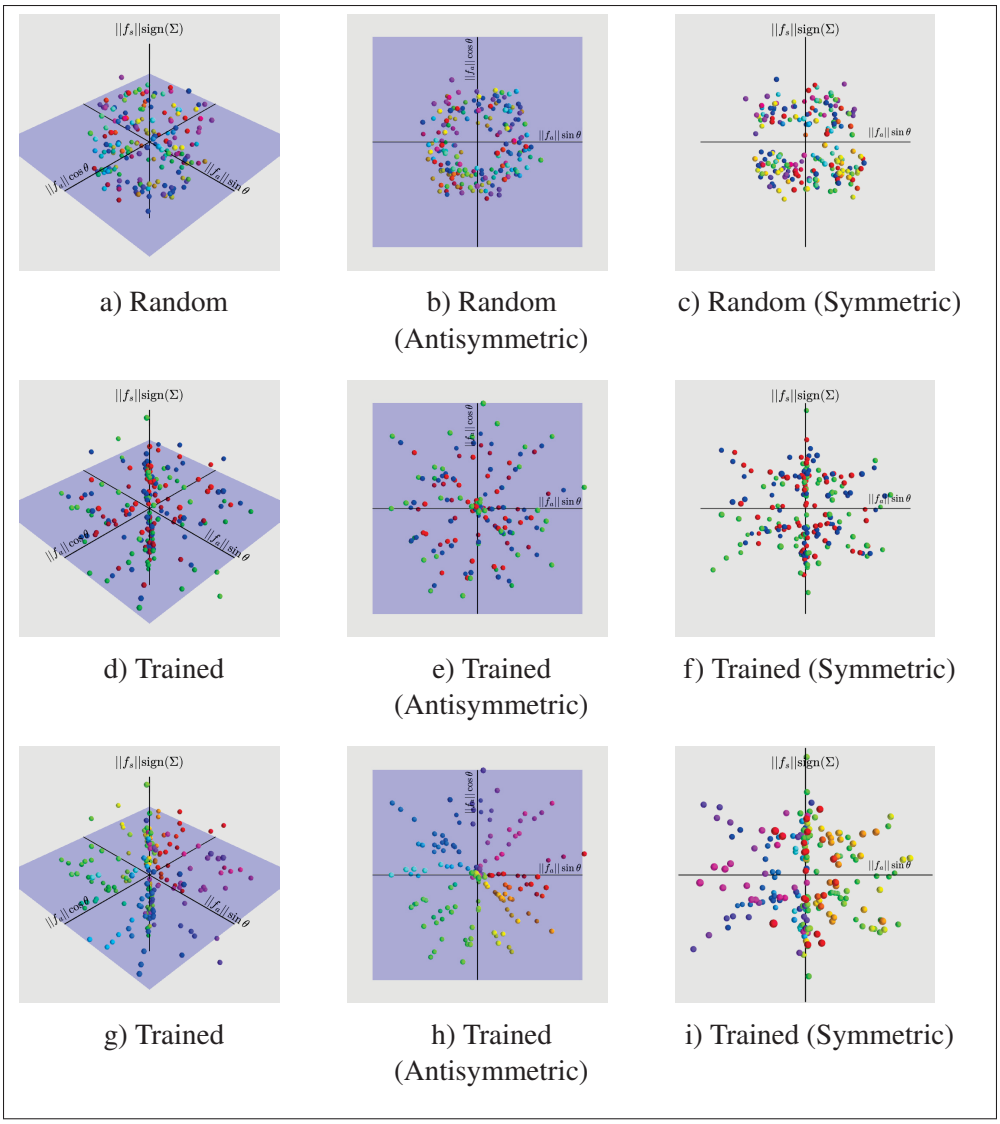


Figure 3.4 All kernels from the input RGB layer of VGG16 trained on ImageNet. (a–c) Randomly initialized kernels, coloured by filters dominant orientation. (d–f) Trained kernels, coloured by RGB channel. (g–i) Trained kernels, coloured by dominant orientation. Trained filters exhibit strong channel correlation (by orientation), with occasional anti-correlated channels.

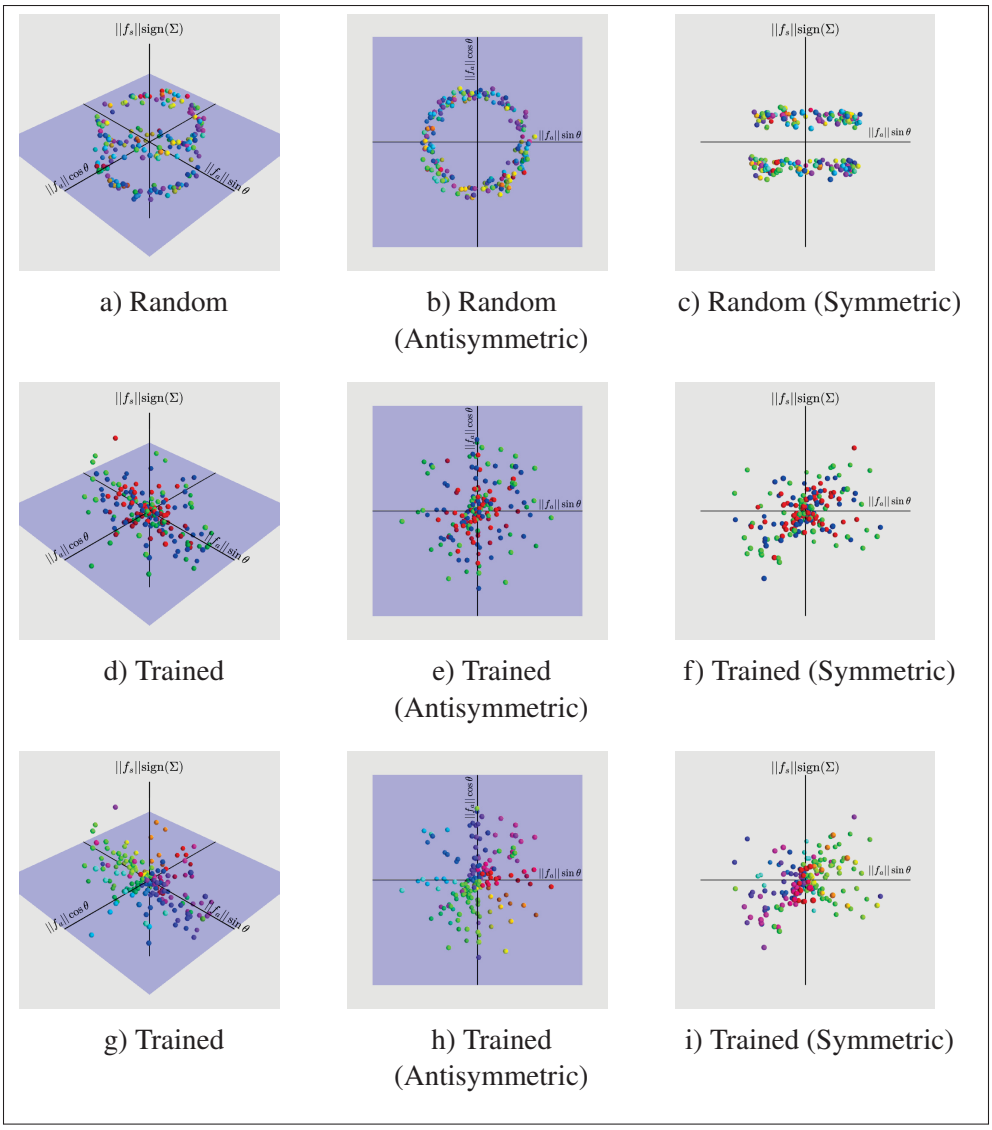


Figure 3.5 All kernels from the input RGB layer of ResNet50 trained on ImageNet. (a–c) Randomly initialized kernels, coloured by filters dominant orientation. (d–f) Trained kernels, coloured by RGB channel. (g–i) Trained kernels, coloured by dominant orientation. Trained filters exhibit strong channel correlation (by orientation), with occasional anti-correlated channels.

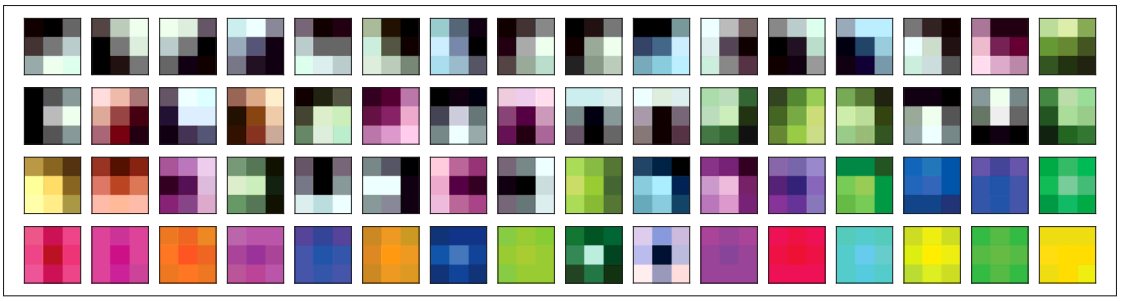


Figure 3.6 Visualization of filters (decreasing β^2 from top left to lower right) from VGG16's RGB layer

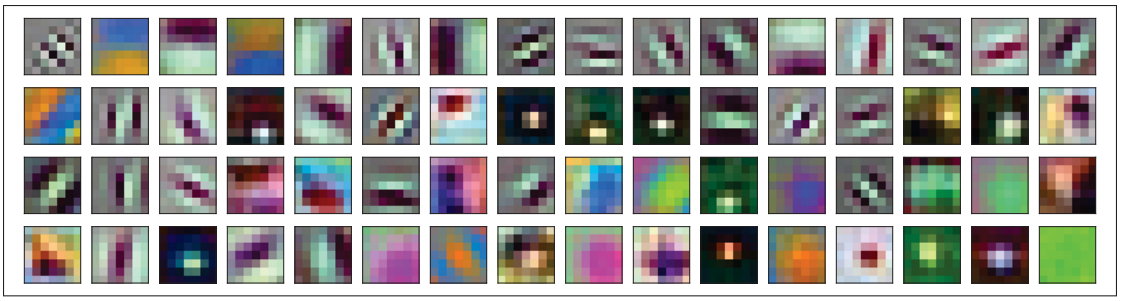
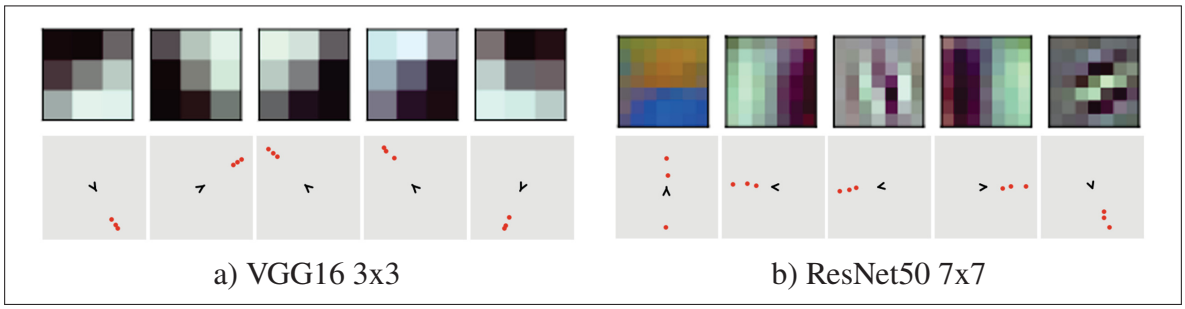


Figure 3.7 Visualization of filters (decreasing β^2 from top left to lower right) from ResNet50's RGB layer



a) VGG16 3x3 b) ResNet50 7x7

Figure 3.8 Visualizing the 5 strongest magnitude input layer RGB filters from trained a) VGG16 and b) ResNet50 networks. The upper row shows filters as 3-channel RGB images, the lower row shows corresponding filters as scatter plots of the odd gradient components $\nabla = \{\nabla_x, \nabla_y\}$ of kernels (red dots). Note that filter channels are typically greyscale and correlated in orientation, anticorrelated channels exhibit colour gradients (yellow, blue) in b)

3.2.2 Trained Kernels Exhibit Antisymmetry or Symmetry

Figure 3.9 shows the distributions of 3×3 kernels for (a) random initialization and throughout layers of (b) VGG16 and (c) ResNet50 trained on ImageNet networks, grouped according to symmetric, antisymmetric or mixed according to ranges of mixing ratio β^2 values. At random initialization in Figure 3.9, β^2 follows a Beta(3, 1.5) distribution, as derived in Section 2.4, which is dominated by mixed components (black) and skewed towards antisymmetric components (blue). After training, individual layers generally exhibit noticeable dominance of either symmetric (red) or antisymmetric (blue) component, where early layers are primarily antisymmetric and deeper layers are primarily symmetric. The differences between VGG and ResNet50 are likely due to architectural differences, e.g. the use of skip connections in ResNet50. It may be interesting to understand specifically how components are linked to network architectural choices; this is left for future work.

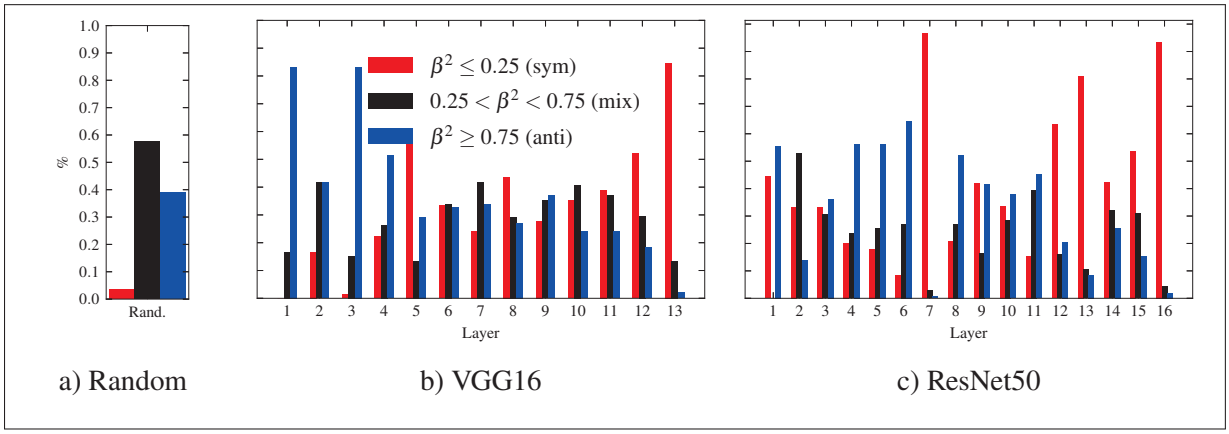


Figure 3.9 Histograms of kernel types in each layer showing the numbers of antisymmetric/symmetric/mixed kernels in each layer according to beta ranges of $\beta^2 = [0 - .25, .25 - .75, .75 - 1]$, and considering solely the top-10% channels of the top-10% filters, as ranked by ℓ^2-norm

3.2.3 Trained Kernels Exhibit a Dominant Bipolar Orientation θ

In this section, we look at the antisymmetric components of trained CNN filters. We observe that trained kernels have antisymmetric components that are correlated in terms of their orientations.

We show how trained filters develop a correlated bipolar orientation distribution. Figure 3.10 shows examples of typical individual multichannel filters in various layers of VGG16 and ResNet50, in the antisymmetric plane (defined by antisymmetric magnitude $||f_a||$ and gradient angle θ). Deep layer filters tend to exhibit bipolar orientation, indicating correlated channels and angular similarity. We provide more examples for VGG16 and ResNet50 in Figures II-3 and II-4.

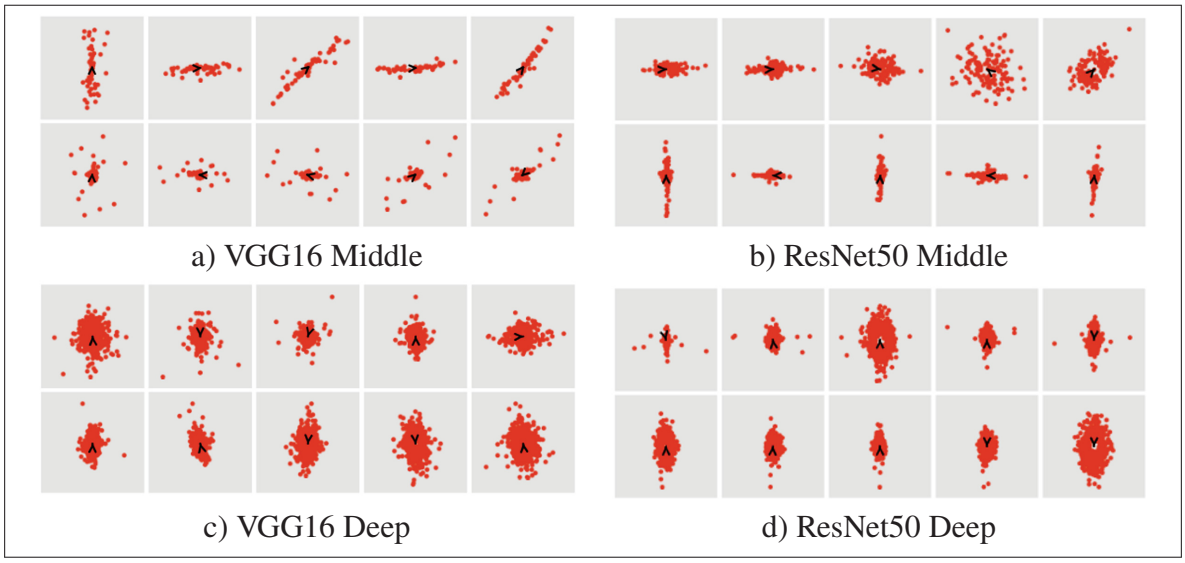


Figure 3.10 Visualizing the 5 strongest magnitude filters from trained VGG16 (right) and ResNet50 (left) networks for middle (upper) and deep (lower) layers. Each filter is shown as a scatter plot of the odd gradient components $\nabla = \{\nabla_x, \nabla_y\}$ of kernels (red dots). Note the bipolar scatter as filters generally consisted of correlated and anti-correlated components about a dominant orientation. Note that deep layer filters are vertical in orientation c) and d)

In Figure 3.10 we observe the antisymmetric plane (defined by antisymmetric magnitude $||f_a||$ and gradient angle θ) in multichannel filters in VGG16 and ResNet50 and note that a bipolar structure along a dominant orientation $\hat{\theta}$. To determine $\hat{\theta}$ of a filter F we calculate the eigenvector of the distribution of F's antisymmetric projection.

Let $\mathbf{r_x}$ and $\mathbf{r_y}$ be vectors of the Cartesian coordinates of each kernel belonging to F, as shown in Figure 2.8.

$$r_x = ||f_a|| \cos(\theta) \tag{3.3}$$

$$r_y = ||f_a|| \sin(\theta) \tag{3.4}$$

By expressing the distribution of kernels in Cartesian coordinates, we can compute the covariance matrix C.

$$C = \begin{bmatrix} \text{Var}(\mathbf{r_x}) & \text{Cov}(\mathbf{r_x}, \mathbf{r_y}) \\ \text{Cov}(\mathbf{r_y}, \mathbf{r_x}) & \text{Var}(\mathbf{r_y}) \end{bmatrix} \tag{3.5}$$

The matrix C can be decomposed as:

$$C = Q \Lambda Q^\top \tag{3.6}$$

where $Q \in \mathbb{R}^{2\times2}$ is an orthonormal matrix of eigenvectors, and $\Lambda = \text{diag}(\lambda_1, \lambda_2)$ contains the corresponding eigenvalues. Let $\mathbf{v_1}$ be the eigenvector corresponding to the largest eigenvalue of C and $\mathbf{v_2}$ be the eigenvector corresponding to its smallest eigenvalue. We then find that $\hat{\theta} = \arg \mathbf{v_1}$.

3.2.4 Trained Kernel Orientations are Correlated Across Layers

Here, we observe how the dominant orientations of trained filters and kernels are correlated between layers. Specifically, kernels tend to operate on channel outputs of similarly oriented filters.

We adopt the notation $F^{\text{Layer}}[\text{channel}, \text{filter}]$ to represent the convolutional weights at a given layer. We begin by measuring the dominant orientation of a filter at a given layer, denoted by $\hat{\theta}_{(F^L[:,i])}$. In a typical CNN architecture, the output of filter $F^{L-1}[:, i]$ is passed to the subsequent convolutional channels $F^L[i, :]$. We compute the angular difference between $\hat{\theta}_{(F^{L-1}[:,i])}$ and the dominant orientation of each receiving channel, $\hat{\theta}_{(F^L[i,:])}$. These angular differences are

discretized and counted. Each count is weighted by the antisymmetric magnitude of the corresponding kernel, $||F_a^L[i,j]||$, relative to the other kernels within the same filter, thus minimizing the orientation contribution of *inactive* neurons. We apply these steps to all filters of all layers in VGG16 (Simonyan & Zisserman, 2015).

Figure 3.11 shows distributions of angular differences between filters and kernels in adjacent layers. We note that although distributions are uniform following random initialization a), a strong correlation may be observed between oriented filters and similarly oriented kernels across all layers following training in b) and c). Furthermore, the distributions follow a cosine-like pattern with peaking at angles $\theta = 0, \pm\pi$, resembling the results in Figure 2.12b, where we demonstrate attenuation in successive convolutions at varying orientations. This suggests that the network organizes itself to preserve coherent orientation information. Not only does the nature of convolution attenuate signals when successive kernels differ in orientation, but the CNN also learns weights that further suppress incoherent orientation signals.

Angular difference histograms across all VGG16 layers may be found in Figure II-7.

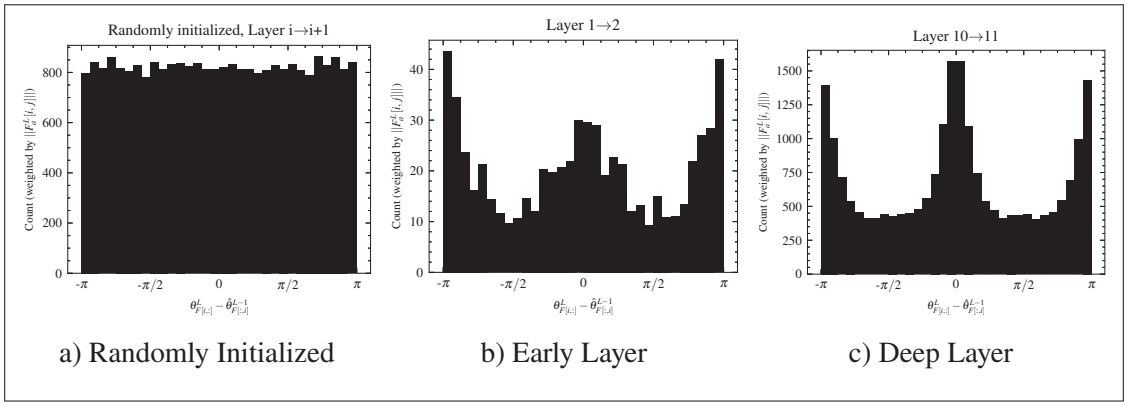


Figure 3.11 Histograms of filter weights associated by channels (kernels in Layer L) to their respective filters (in Layer $L-1$) according to their gradient angular differences $(\theta_{F^L[i,:]} - \hat{\theta}_{F^{L-1}[:,i]})$ in VGG16

CHAPTER 4

DISCUSSION

In this work, we introduced a new perspective on convolutional neural network (CNN) filtering by decomposing filters into symmetric and antisymmetric components, with the goal of understanding how these operators influence information propagation and the expansion of the effective receptive field in trained networks. This thesis describes the geometrical properties of these symmetric and antisymmetric components, and their unique mechanical effects upon activation images under rectified convolution operations commonly applied in convolutional neural networks.

Our approach consists of studying the frequency components (DCT bases) which can be found in 3×3 convolutional kernels. Through simulations of deep convolutional sequences with various bases, we observed that antisymmetric bases induce directional, momentum-like propagation, while symmetric bases consistently produce isotropic diffusion of image content. Examination of trained CNN filters revealed that low-order DCT components, specifically the Σ (symmetric DC) and ∇ (antisymmetric first-order gradient) bases, dominate learned filters, justifying our focus on these components. This finding could lead to lower parameter (compressed) representations of CNNs. Furthermore, we found that applying ReLU between convolutions with ∇ enables a maximal propagation of information in image space.

We further investigated the orientation-dependent behaviour of antisymmetric filters under sequential convolutions for various antisymmetric frequency components. When an image is convolved with a filter and then with its rotated version, maximal signal attenuation occurs at a rotation of $\pm\frac{\pi}{2}$ radians. Experiments on test patterns (*i.e.* random noise, impulse, circle) confirm that information is attenuated according to the angular difference between the filter's orientation and image content. Optimal transmission follows bi-directional gradient filtering, resulting in a random walk where information either oscillates in place or translates up to a maximal velocity.

Empirical validation supports this hypothesis; constraining CNN training to only the DC and gradient components of the filter spectrum retains over 90% of ImageNet classification accuracy in both VGG and ResNet architectures. This demonstrates that small CNN kernels are functionally dominated by symmetric diffusion (DC) and antisymmetric gradient operators ∇_x, ∇_y associated with information vibration or translation resulting from a bi-directional random walk.

The antisymmetric component is closely tied to a dominant orientation θ in the 2D image plane, as defined by the orthogonal gradient bases ∇_x and ∇_y. Deep layer filters tend to exhibit bipolar orientation, indicating correlated channels and angular similarity $\hat{\theta}$. Similar bipolarity in convolutional filters has also been observed by Cammarata *et al.* (2020), who, using tuning curves, found that filters maximized for a given orientation $\hat{\theta}$ also respond to stimuli rotated by π. This result is consistent with research on early filters in the mammalian visual system (Hubel & Wiesel, 1962). We observe that the final layers of the network seem to converge to a dominant orientation of $\frac{\pi}{2}\ rad$. This observation has also been noted by Petrov *et al.* (2021), who describe the phenomenon as *weight banding*. Petrov *et al.* (2021) find that this phenomenon is due to the structural properties of ImageNet; when trained on images rotated by $\frac{\pi}{2}\ rad$, the banding becomes horizontal (Petrov *et al.*, 2021).

Analysis of trained VGG16 models reveals that filters tend to organize along these dominant orientations, with alignment persisting across layers through strong correlations between filter-channel orientations. This organization mirrors the theoretical attenuation behaviour of sequential convolutions and suggests that CNNs implicitly learn to preserve angular coherent information aligned with each filter's dominant direction while suppressing information of misaligned channels. Beyond the intrinsic attenuation introduced by convolution when successive kernels differ in orientation, the network also develops a weight structure that further suppresses information propagating through orthogonal channels. Our current analysis is limited to two consecutive 3×3 layers, which is why we focus on VGG16 (Simonyan & Zisserman, 2015); more complex architectures such as ResNet (He *et al.*, 2016) do not always exhibit direct filter-channel

connections. While preliminary, our results encourage future exploration across other networks to better understand the emergence of complex learned structures.

A number of future directions exist. This work was limited to the interpretation of learned kernels, as well as a study of propagation properties intrinsic to convolution, but several promising directions remain. One direction we aim to pursue is the effect of orientation on the Effective Receptive Field (ERF) of a trained CNN (Luo *et al.*, 2016). It has been empirically and analytically observed that the ERF of a randomly initialized CNN grows linearly with depth n at a rate of $O(\sqrt{n})$, though, relative to the theoretical receptive field, it shrinks at a rate of $O(1/\sqrt{n})$ (Luo *et al.*, 2016). After training, however, Luo *et al.* (2016) observe that the ERF typically grows to match the theoretical receptive field, though the underlying cause remains unclear. Based on our findings, we posit that the coherent orientation structures shown in Figure II-7 actually allow for image propagation (as demonstrated in Figure 2.9b and Figure 2.9c), which may explain how the ERF expands post-training. Although studying this in detail lies beyond the scope of this thesis, we aim to investigate it in future work.

Code for generating the results in this thesis may be found at [1].

[1] https://github.com/liamaltarac/Information-Mechanics

CONCLUSION AND RECOMMENDATIONS

This thesis presented a new framework for understanding convolutional neural networks (CNNs) through the geometric decomposition of convolutional filters into symmetric and antisymmetric components. Spectral analysis reveals that small convolutional kernels are dominated by low-order frequency bases, specifically the DC and gradient DCT bases, which correspond to distinct modes of information propagation. Symmetric filters induce isotropic diffusion, while antisymmetric filters generate directional, momentum-like translation that propagates at maximal velocity under ReLU. Together, these components define the mechanical behaviour of rectified convolution, providing an interpretable link between kernel geometry, information flow, and the expansion of the effective receptive field.

Empirical results on VGG16 and ResNet50 confirmed that low-frequency (Σ and ∇) components alone preserve over 92% of ImageNet classification accuracy. Furthermore, we observed that filters self-organize into coherent orientation structures that persist across layers, suggesting that CNNs implicitly learn to maintain angular consistency while suppressing orthogonal activations.

Future work should explore how orientation coherence contributes to the growth of the effective receptive field during training, as well as extend the proposed framework to non-convolutional and transformer architectures. Investigating frequency-aware initialization schemes and incorporating symmetry constraints could further improve training stability, interpretability in terms of the forward propagation mechanism, and generalizability in modern neural networks. Future research may also consider formulating CNNs under a reduced-parameter framework based on the symmetric (Σ) and antisymmetric (∇) components, potentially yielding more compact and mechanistically interpretable models. Moreover, improved learning algorithms could be developed by explicitly accounting for symmetric and antisymmetric information, *e.g.* during initialization, gradient backpropagation, or potentially entirely new methods with forward propagation.

APPENDIX I

CHAPTER 2 – ADDITIONAL RESULTS

I.1 Examples of Propagation

Here we demonstrate the results of rectified convolution from a single channel, and how the velocity of information is determined by the mixing ratio β of even and odd filter components, similarly to the Lorentz transform in the theory of special relativity. Rectified convolution is repeatedly performed upon test patterns, and the velocity is measured in terms of the displacement of the centre of mass per convolution.

I.1.1 Experimental Setup

Here we demonstrate the mechanics by which even (e.g. DC Σ) and odd (e.g. gradient ∇_x, ∇_y) filter components act upon image information, similarly to Section 2.3 of the paper, for various combinations of test patterns (pixel, circle), filter sizes (2×2, 3×3), types (DC, gradient, translation) and activation functions (none, ReLU, Modulus).

Table I-1 shows the filter kernels used for various values of β^2. Most demonstrations mix Σ and ∇_x components according to the β^2 parameter, and convolve a test pattern. Note that 2x2 kernels are applied alternatingly within a 3x3 kernel in order to avoid a half-pixel shift following convolution. We also test a special case of propagation with a translation kernel, which is normally an offset impulse kernel (Table I-1, 3×3 translation for $\beta^2 = 0.75$).

Our demonstrations perform rectified convolution on two test image patterns including a circle ($r = 19$) fig. I-1a) and an impulse (fig. I-1b). Between each iteration, the activation centre of mass μ_x and standard deviation σ_x are computed from a normalized activation $f(x, y)$ as follows:

$$\mu_x = \frac{\sum_x x \|f(x,0)\|}{\sum_x \|f(x,0)\|} \qquad \sigma^2 = \frac{\sum_x \|f(x,0)\|(x-\mu_x)^2}{\sum_x \|f(x,0)\|} \tag{A I-1}$$

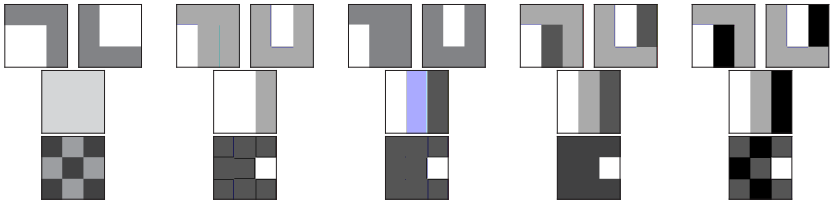
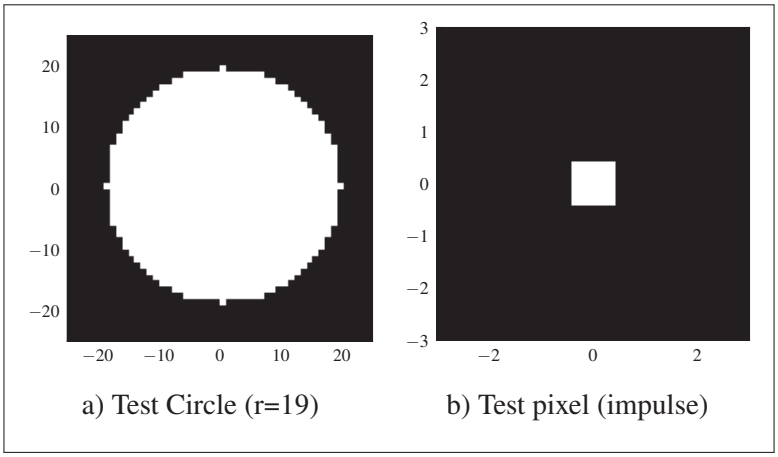
Kernel size	β^2				
	0	0.25	0.5	0.75	1
2×2 (alternating)					
3×3					
3×3 (translation)					

Table-A I-1 Kernel Examples. Generating kernel f according to mixing ratio β .

$$f = \beta \hat{f}_a + \sqrt{1-\beta^2}\,\hat{f}_s$$

,



a) Test Circle (r=19) b) Test pixel (impulse)

Figure-A I-1 Test Patterns

I.1.2 Convolution Without Activation

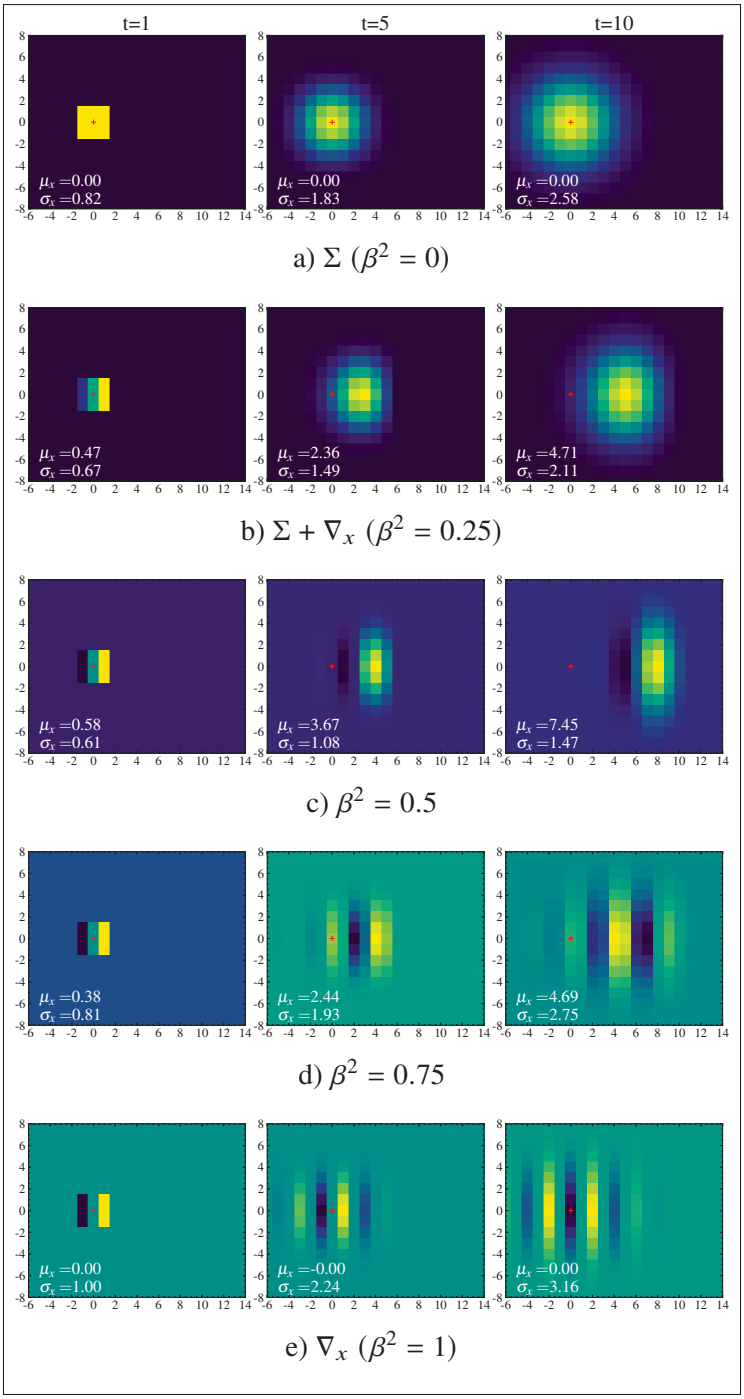


Figure-A I-2 Demonstrating the effect of repeated convolution (no activation function) of a test pattern (pixel) over 3×3 kernels, varying β mixing between Σ and ∇. Note that for both $\beta = 0$ and $\beta = 1$, there is net displacement of the centre of mass, this is distinctly different from rectified convolution

I.1.3 Convolution With ReLU

I.1.3.1 Unipolar, 3×3 kernel

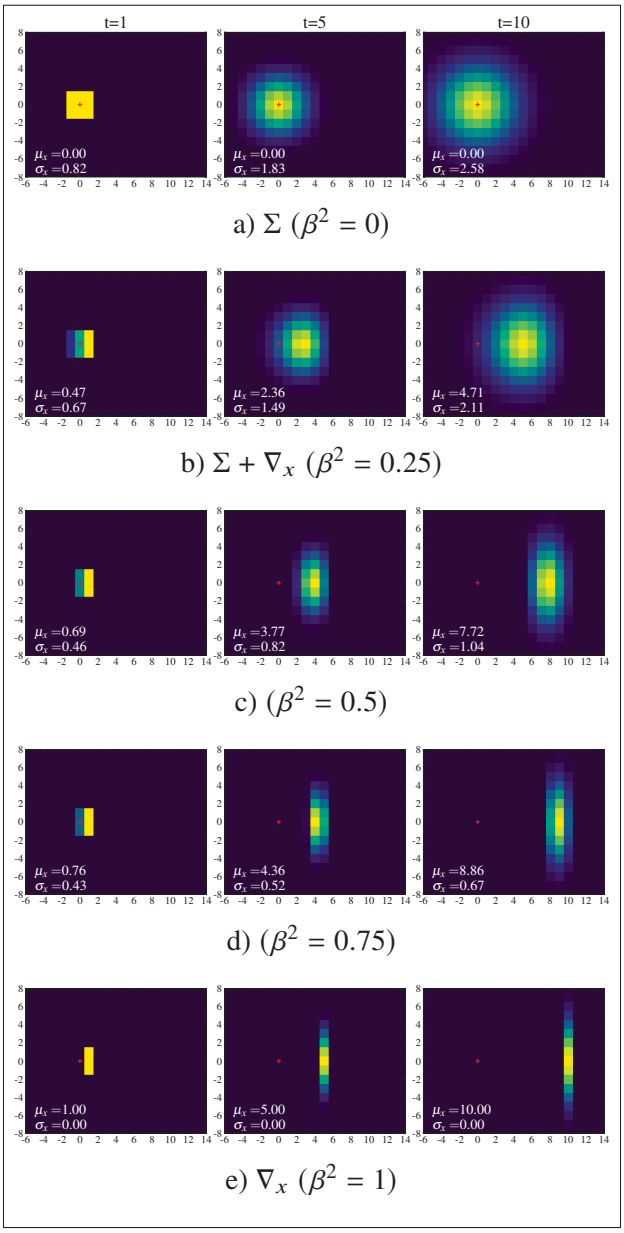


Figure-A I-3 Demonstrating the effect of repeated convolution+ReLU of a test pattern over different types of 3×3 kernels (DC and Gradient). Note that for $\beta = 0$ a), content diffuses symmetrically about a stationary centre of mass, while for $\beta = 1$ the centre of mass translates rightward with maximum velocity

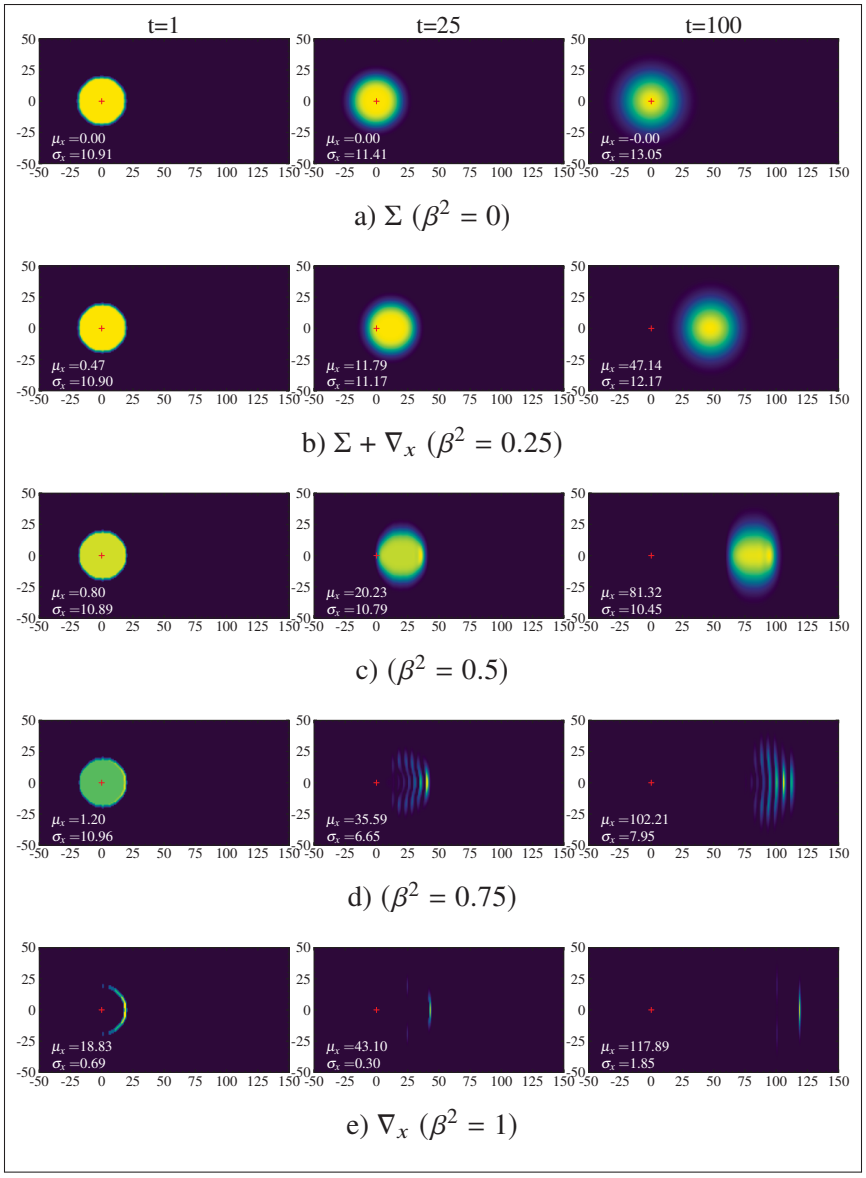


Figure-A I-4 Demonstrating the effect of repeated convolution+ReLU of a circular test pattern ($r = 19$) with 3×3 kernels mixing DC Σ and fixed direction gradient ∇_x over various mixing ratios $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$. Note that for $\beta = 0$ a), content diffuses symmetrically about a stationary centre of mass, while for $\beta = 1$, the circle bulk disappears and the rightmost edge translates right with maximum velocity.

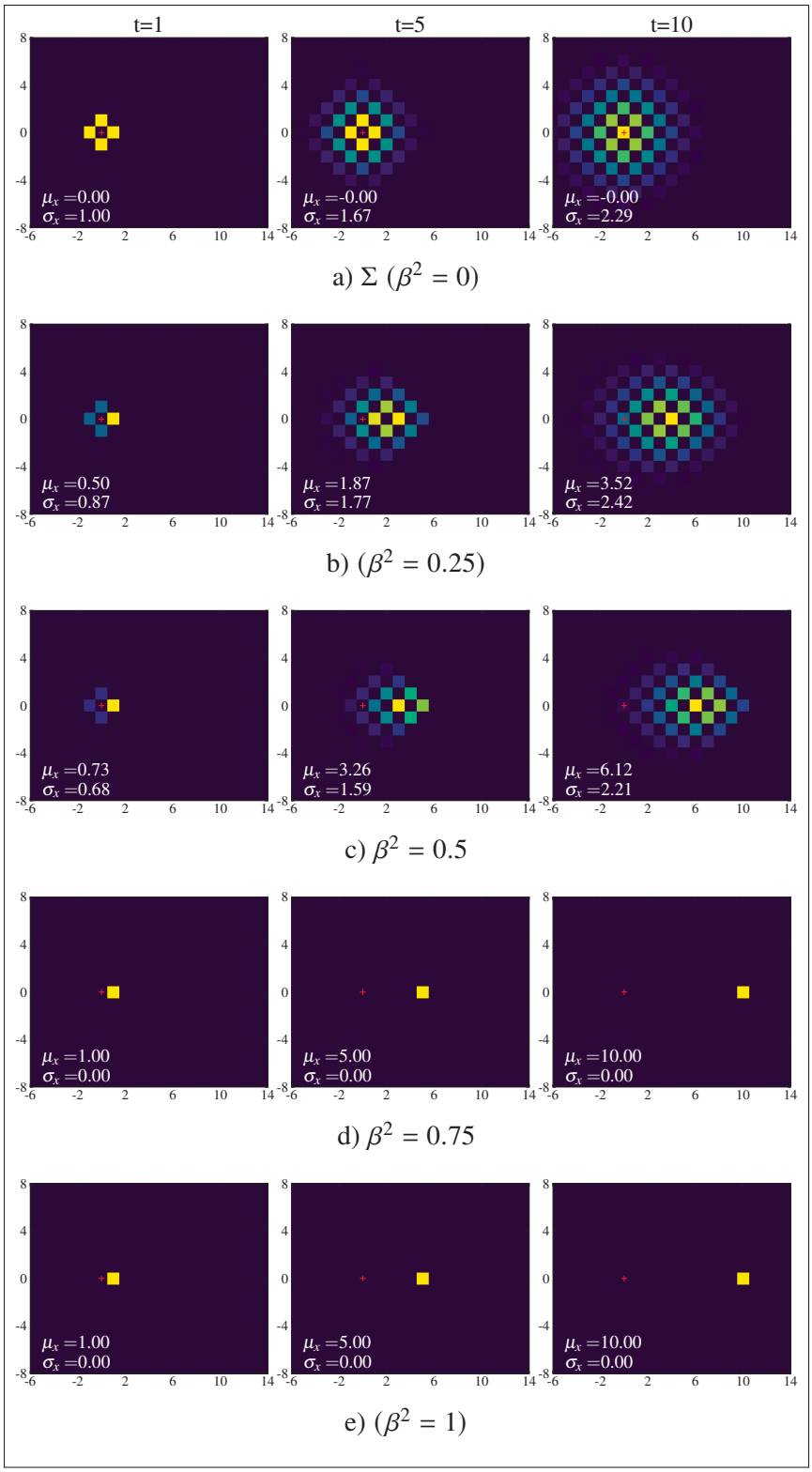


Figure-A I-5 Demonstrating the effect of repeated convolution+ReLU of a test pattern (pixel) over 3×3 translations, varying β. Note that for $\beta = 0$ a), artificial checker-board structure appears due to the complex non-DC (symmetric) component, while the content maintains a stationary centre of mass. For $\beta = 1$, the content translates rightward with maximum velocity.

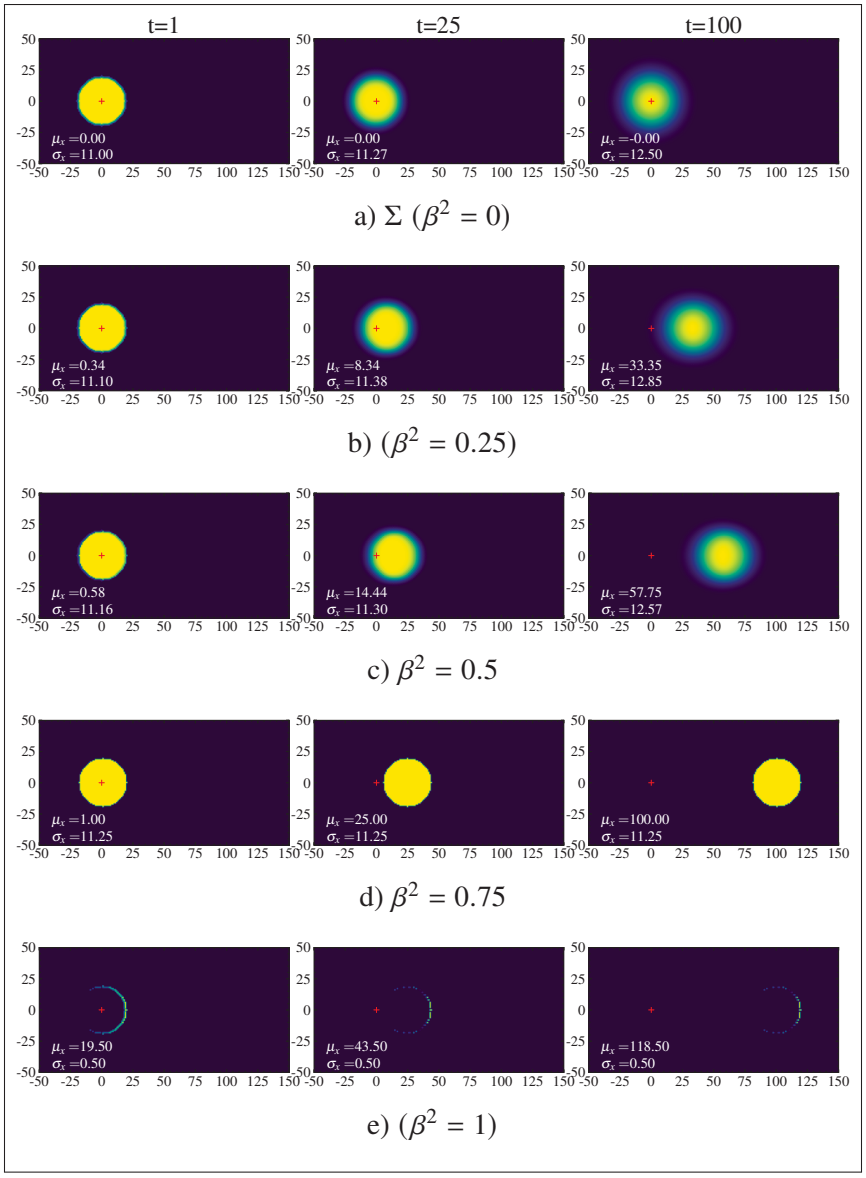


Figure-A I-6 Demonstrating the effect of repeated convolution+ReLU of a test pattern (circle) over 3×3 translation filter components, varying β. Note that for $\beta = 1$, the circle bulk disappears and the rightmost edge translates rightward with maximum velocity.

I.1.3.2 Bipolar, 3×3 kernel

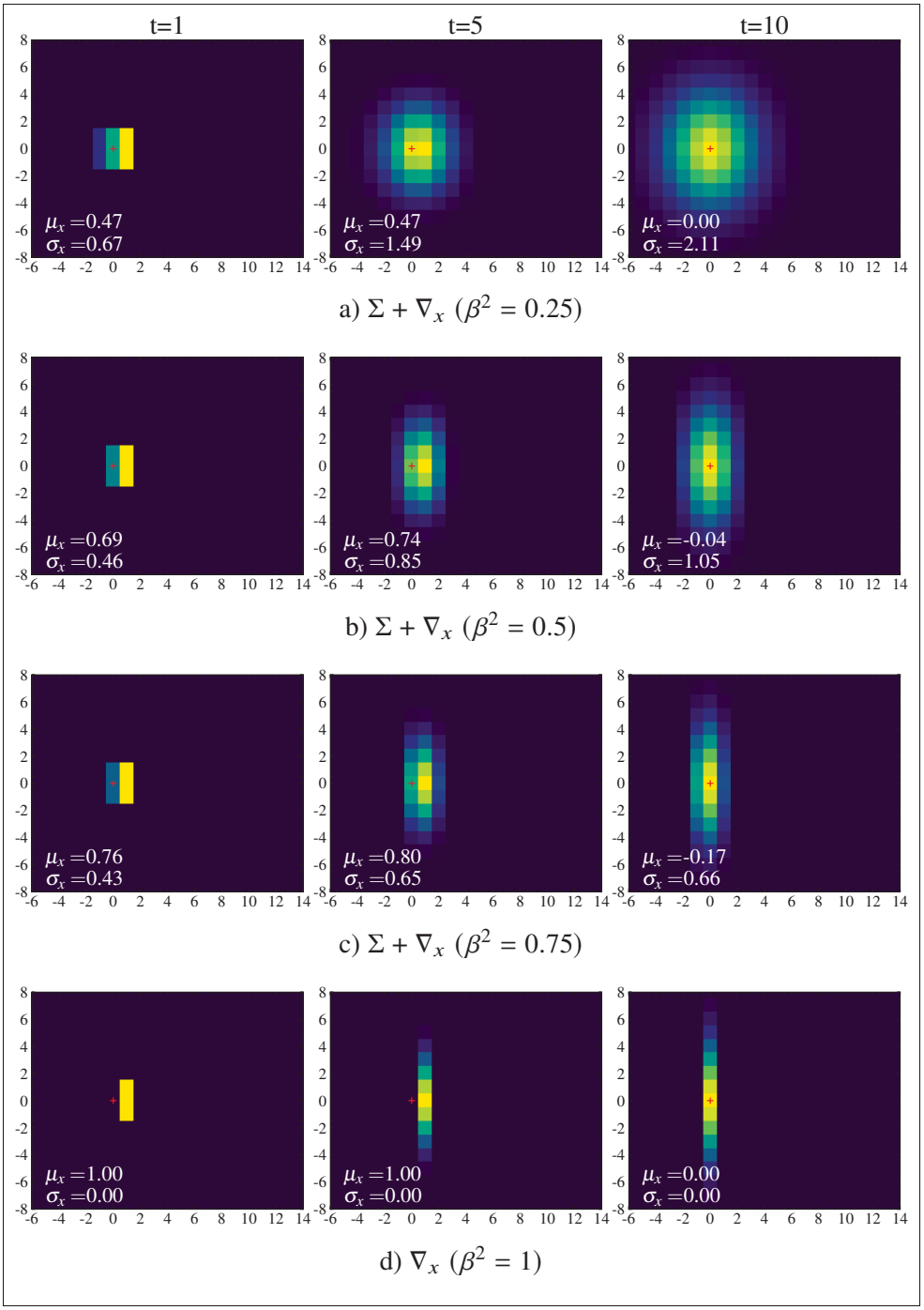


Figure-A I-7 Demonstrating the effect of repeated convolution+ReLU with alternating orientation. Note that for $\beta = 1$, content vibrates left and right, and there is no net translation of the centre of mass.

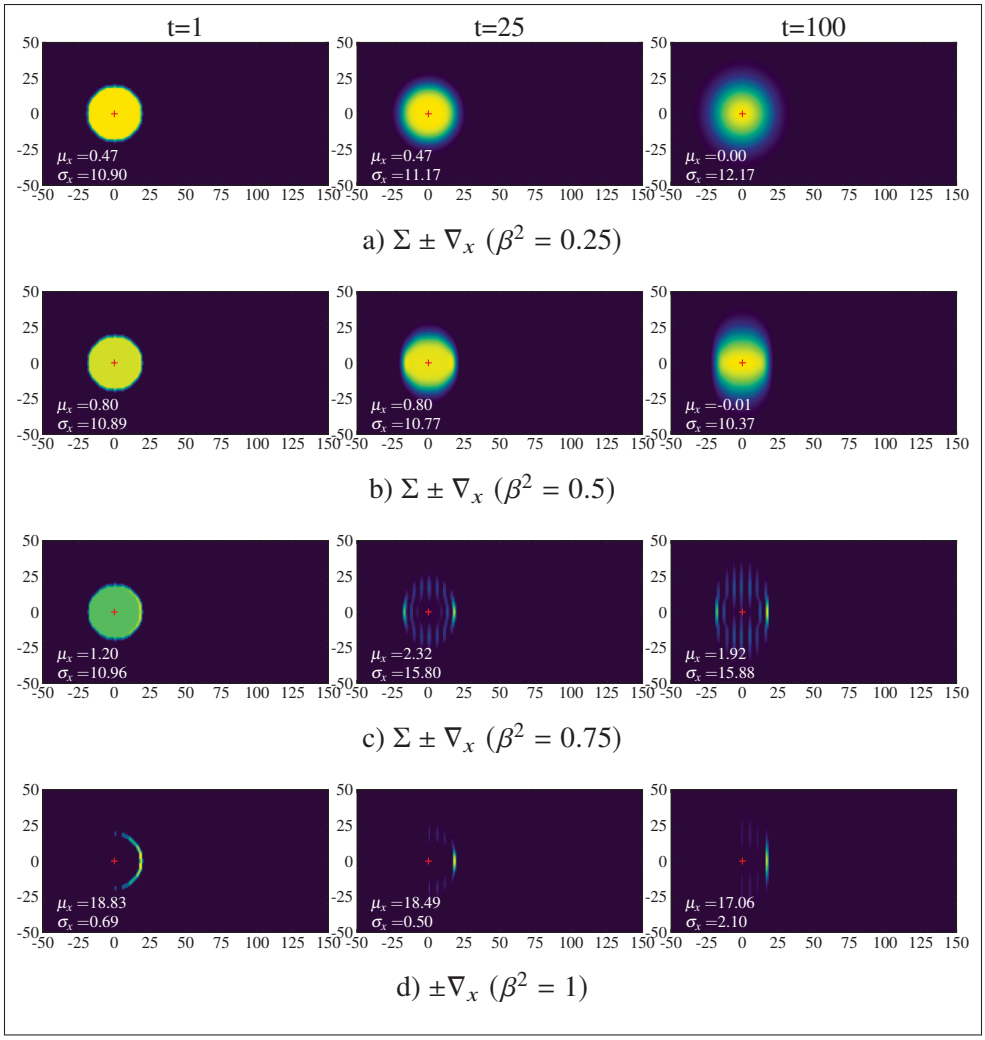


Figure-A I-8 Demonstrating the effect of repeated convolution+ReLU of a circular test pattern over different types of 3×3 kernels mixing DC Σ and alternating direction gradient $\pm\nabla_x$ components for different mixing ratios. Note that for $\beta = 1$, the circle bulk disappears, and the right edge of the circle vibrates left to right with no net translation.

I.1.3.3 Unipolar, 2×2 kernel

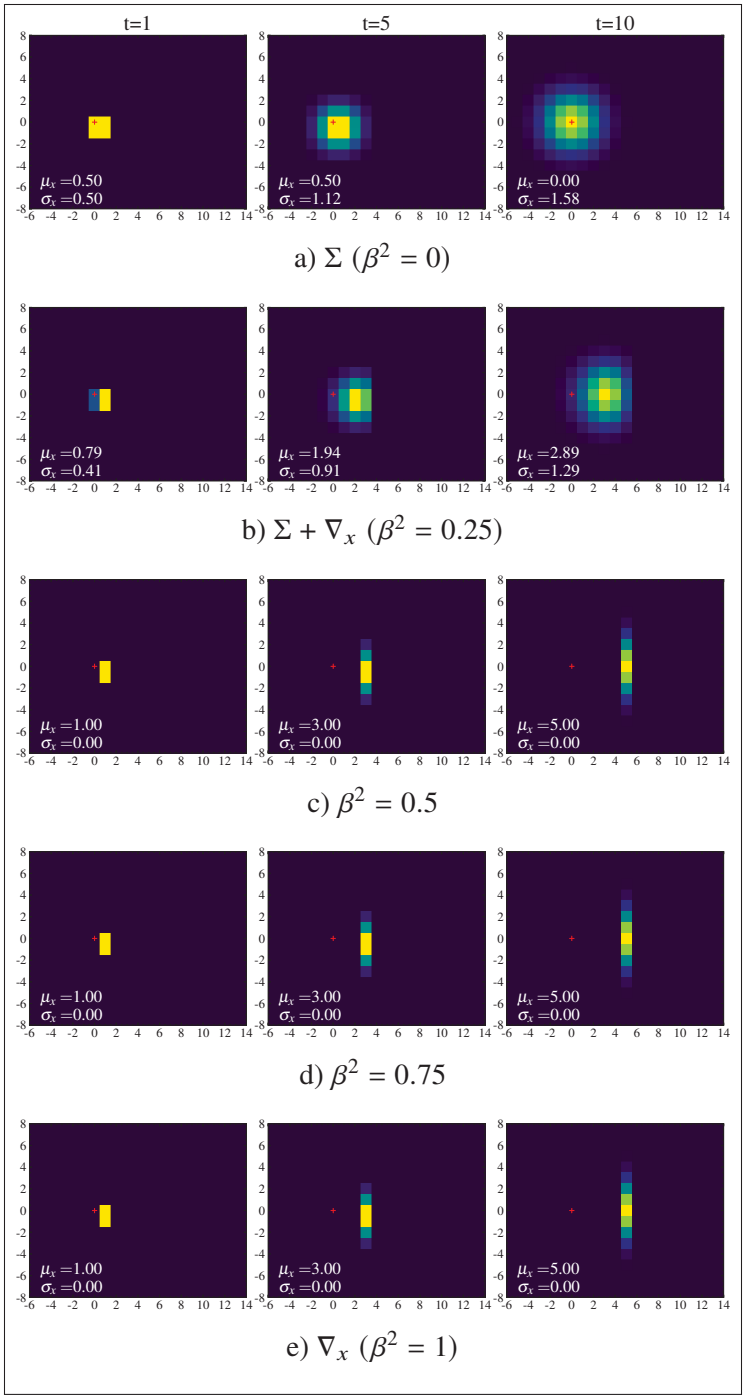


Figure-A I-9 Demonstrating the effect of repeated convolution+ReLU of a test pattern over different types of 2×2 kernels (DC and Gradient). Note that for $\beta^2 \geq 0.5$, the content centre of mass travels rightward with maximum velocity.

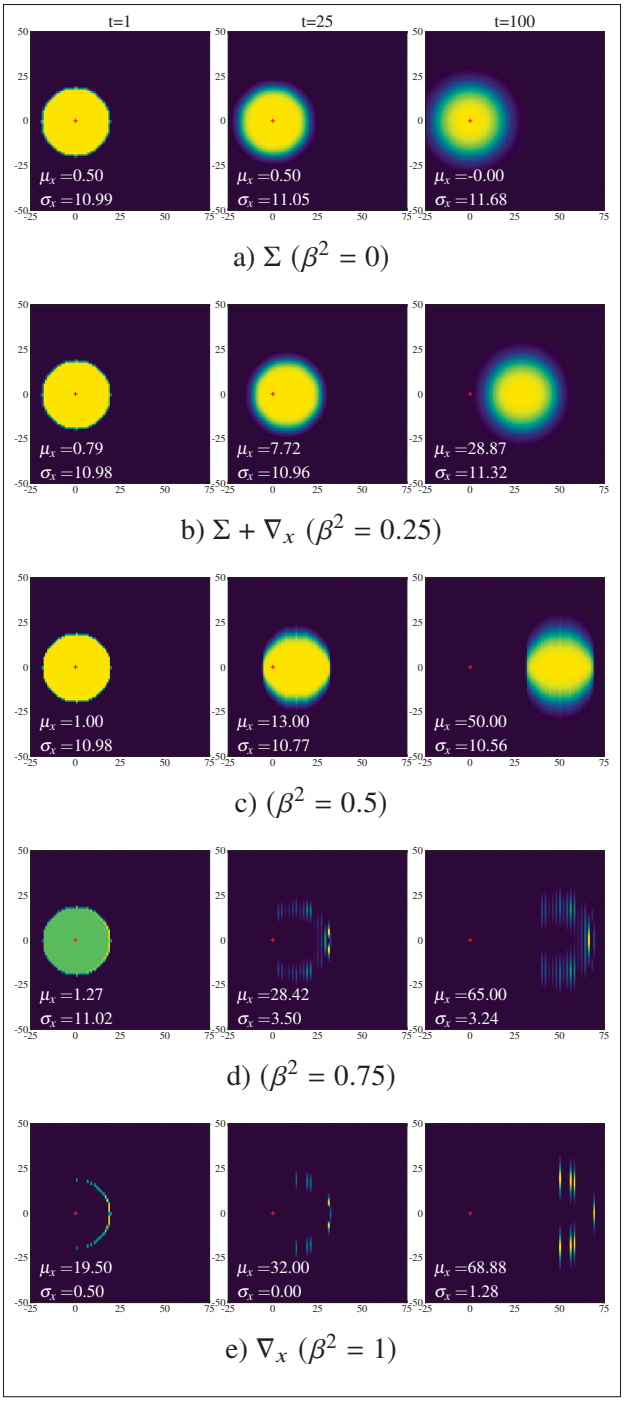


Figure-A I-10 Demonstrating the effect of repeated convolution+ReLU of a circular test pattern ($r = 19$) with 2×2 kernels mixing DC Σ and fixed direction gradient ∇_x components over various mixing ratios β. Note that for $\beta = 0$, the content diffuses symmetrically with a stationary centre of mass, while for $\beta = 1$ the circle bulk disappears and the right edge of the circle travels rightward with maximum velocity.

I.1.4 Convolution With Mod (Absolute value)

I.1.4.1 Unipolar, 3×3 kernel

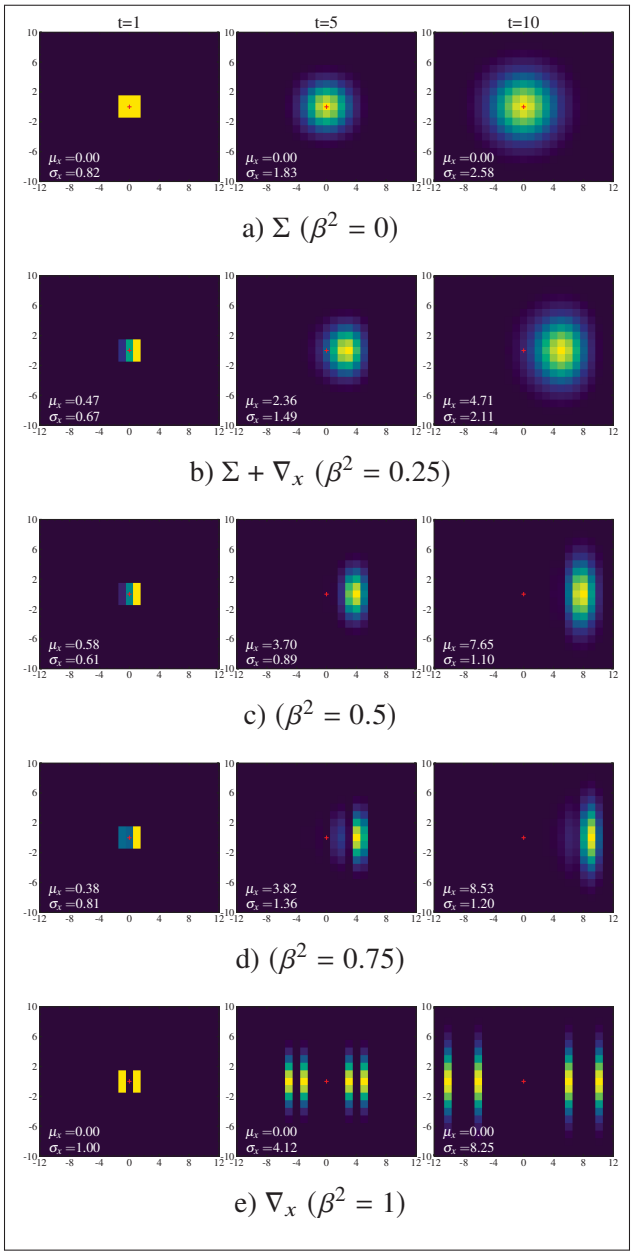


Figure-A I-11 Demonstrating the effect of repeated **convolution+mod** of a test pattern over different types of 3×3 kernels (DC and Gradient). Note that for $\beta = 0$, the content diffuses symmetrically with a stationary centre of mass, while for $\beta = 1$ the pattern propagates symmetrically in both directions at maximum velocity with a stationary centre of mass.

I.2 Sequential Convolution At Varying Orientations

I.2.1 Experimental Setup

The experiments consist of demonstrating the effect of sequential convolution with antisymmetric kernels, which we rotate according to an angle θ. In order to generate rotated kernels, we interpolate between two orthogonal quadrature pair bases, a method akin to steerable filters (Freeman *et al.*, 1991).

For the purpose of this thesis, we limit ourselves to the study of 3×3 kernels, and therefore the 3 following antisymmetric kernels Figure I-12 : Gradient ∇ ($D_{0,1}$), Saddle $D_{1,1}$ and $D_{1,2}$.

Steering the gradient basis is straight out of Freeman *et al.* (1991). See Equation (A I-2) and Figure I-12 for examples of steered gradient bases at various orientations .

$$F_\theta = \nabla_x \cos\theta + \nabla_y \sin\theta \tag{A I-2}$$

Steering the saddle basis $D_{1,1}$ is done in a manner slightly different from Freeman *et al.* (1991), however we still do an interpolation between quadrature pairs : $D_{1,1}$ and $D_{0,2} - D_{2,0}$. While Freeman *et al.* (1991) aims to steer $D_{0,2}$ (in actuality its analog as a Gaussian derivative), which contains a symmetric component, we seek to steer $D_{1,1}$, which Freeman *et al.* (1991) does not specify. We, however, find that $D_{0,2} - D_{2,0}$ actually removes the symmetric component, thus resulting in a saddle kernel orthogonal to $D_{1,1}$. We steer $D_{1,1}$ according to Equation (A I-3). See Figure I-13 for examples of steering $D_{1,1}$.

$$F_\theta = (D_{2,0} - D_{0,2}) \cos 2\theta + D_{1,1} \ \sin 2\theta \tag{A I-3}$$

Steering the basis $D_{1,2}$ is done according to Equation (A I-4). See Figure I-14 for examples of steering $D_{1,2}$.

$$F_\theta = D_{1,2} \cos\theta + D_{2,1} \sin\theta \tag{A I-4}$$

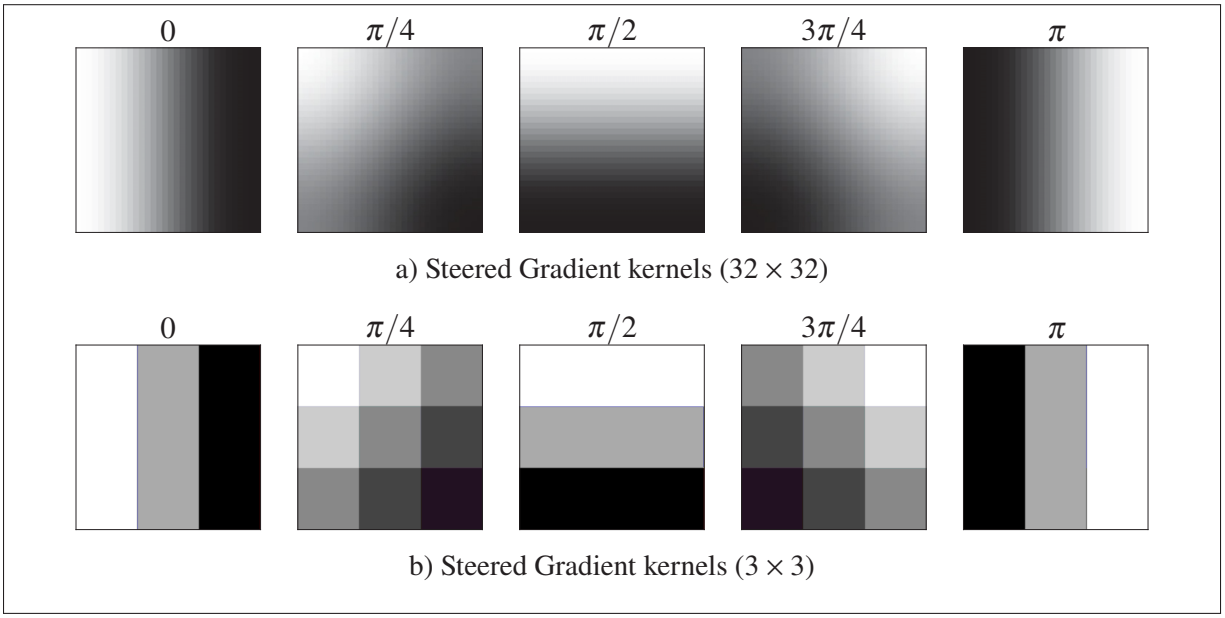


Figure-A I-12 Examples of steered kernels according to $\nabla_\theta = \nabla_x \cos\theta + \nabla_y \sin\theta$ as described by Freeman *et al.* (1991)

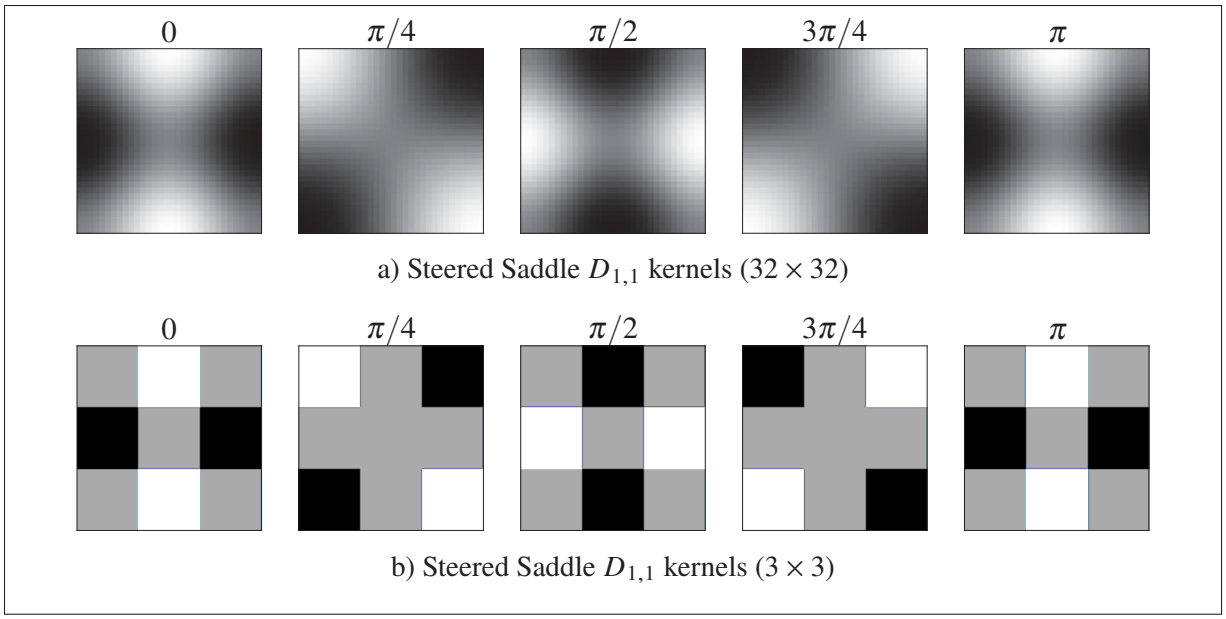


Figure-A I-13 Examples of steered kernels according to $F_\theta = (D_{2,0} - D_{0,2}) \cos 2\theta + D_{1,1} \sin 2\theta$

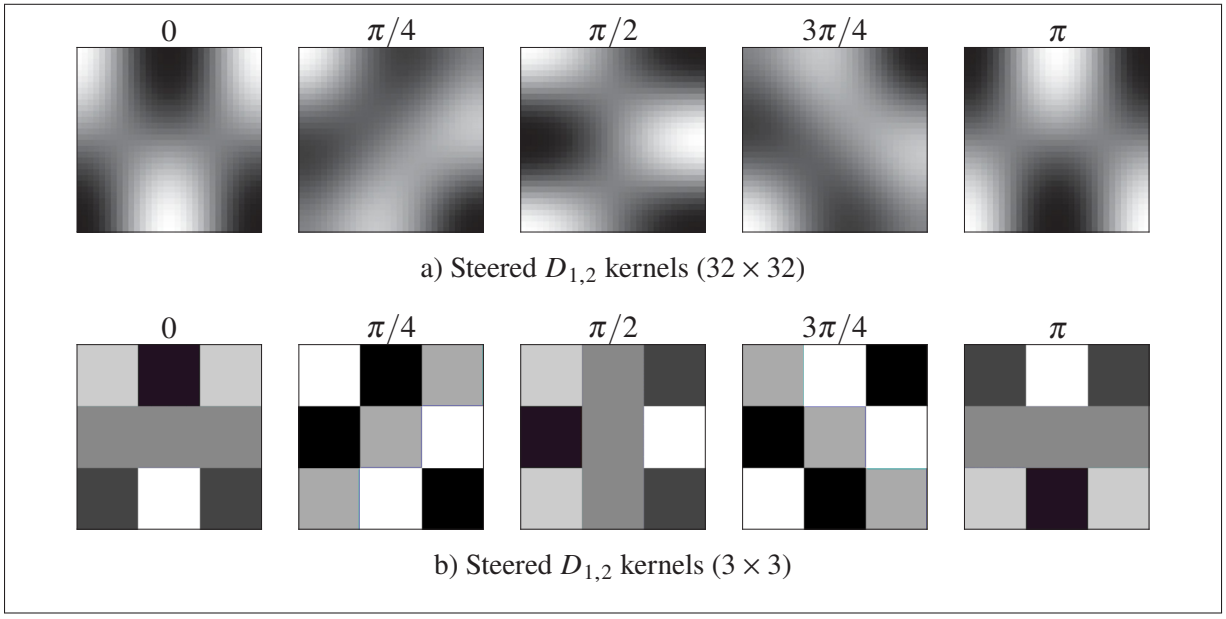


Figure-A I-14 Examples of steered kernels according to $F_\theta = D_{1,2} \cos \theta + D_{2,1} \sin \theta$

I.2.2 Sequential Convolution with Gradient Kernels ∇ at Varying Orientations

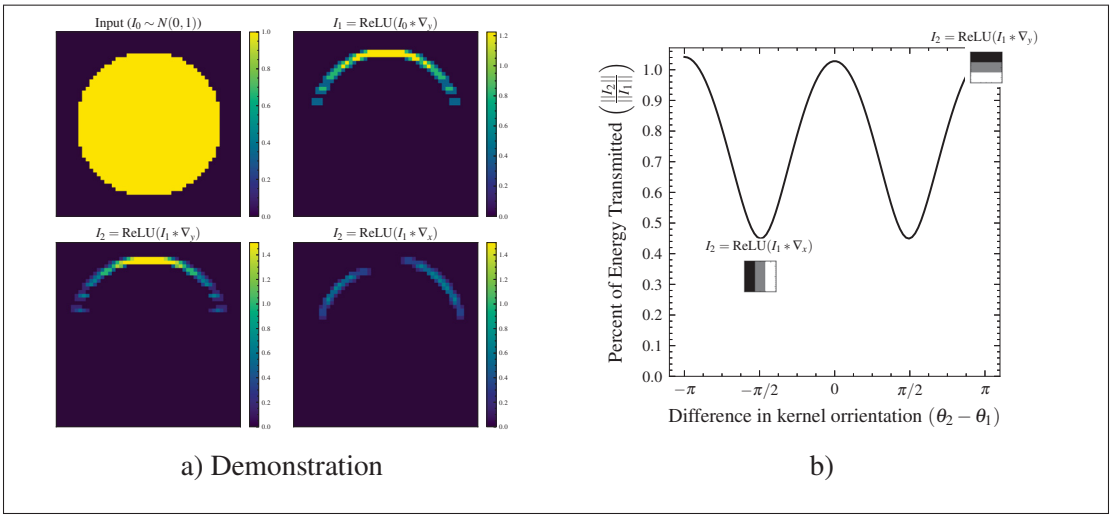


Figure-A I-15 (a) Sequential convolution with identical antisymmetric (gradient) filters at varying relative orientations. Activation magnitude after applying two gradient filters, ∇_1 and ∇_2, with angular difference $\theta_1 - \theta_2$, to a circular test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $\nabla_2 = \nabla_x \cos\theta_2 + \nabla_y \sin\theta_2$. Attenuation is maximal at an angular difference of $\pi/2$.

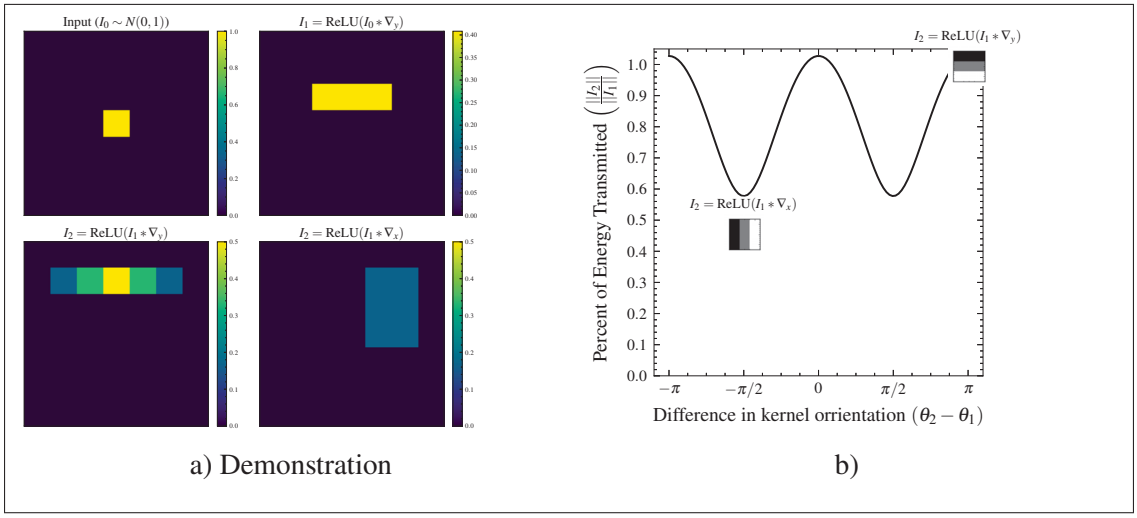
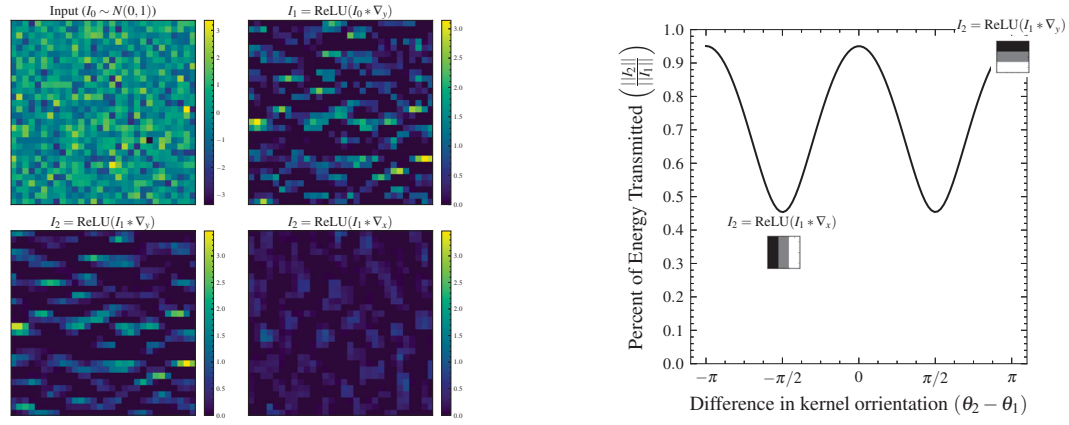


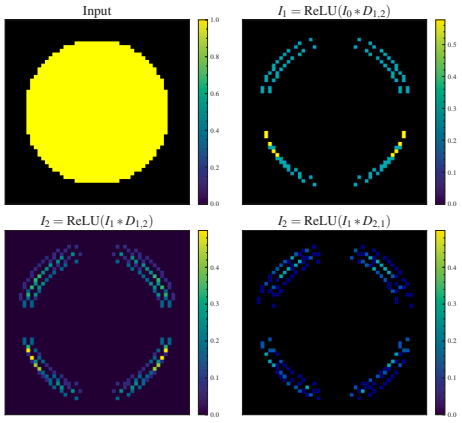
Figure-A I-16 (a) Sequential convolution with identical antisymmetric (gradient) filters at varying relative orientations. Activation magnitude after applying two gradient filters, ∇_1 and ∇_2, with angular difference $\theta_1 - \theta_2$, to a pixel test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $\nabla_2 = \nabla_x \cos\theta_2 + \nabla_y \sin\theta_2$. Attenuation is maximal at an angular difference of $\pi/2$.



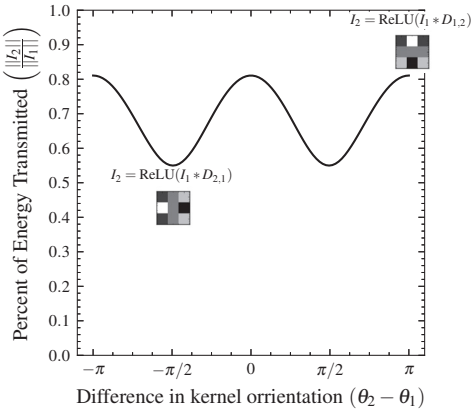
a) Demonstration b)

Figure-A I-17 (a) Sequential convolution with identical antisymmetric (gradient) filters at varying relative orientations. Activation magnitude after applying two gradient filters, ∇_1 and ∇_2, with angular difference $\theta_1 - \theta_2$, to a random test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $\nabla_2 = \nabla_x \cos\theta_2 + \nabla_y \sin\theta_2$. Attenuation is maximal at an angular difference of $\pi/2$.

I.2.3 Sequential Convolution with $D_{1,2}$ at Varying Orientations

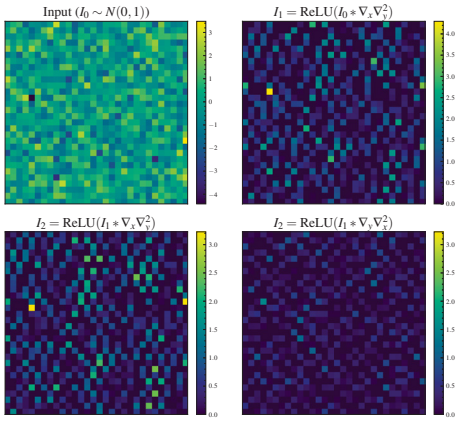


a) Demonstration

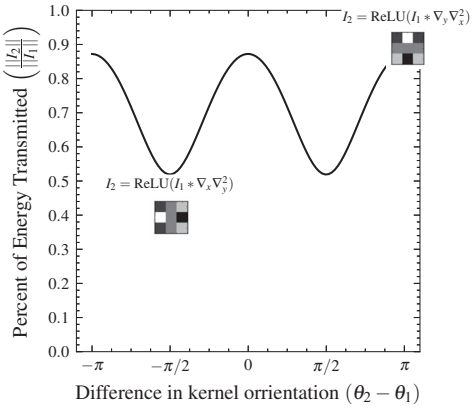


b)

Figure-A I-18 (a) Sequential convolution with identical antisymmetric ($D_{1,2}$,$D_{2,1}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a circular test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = D_{1,2} \cos\theta + D_{2,1} \sin\theta$. Attenuation is maximal at an angular difference of $\pi/2$.



a) Demonstration



b)

Figure-A I-19 (a) Sequential convolution with identical antisymmetric ($D_{1,2}$ and $D_{2,1}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a random test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = D_{1,2} \cos\theta + D_{2,1} \sin\theta$. Attenuation is maximal at an angular difference of $\pi/2$.

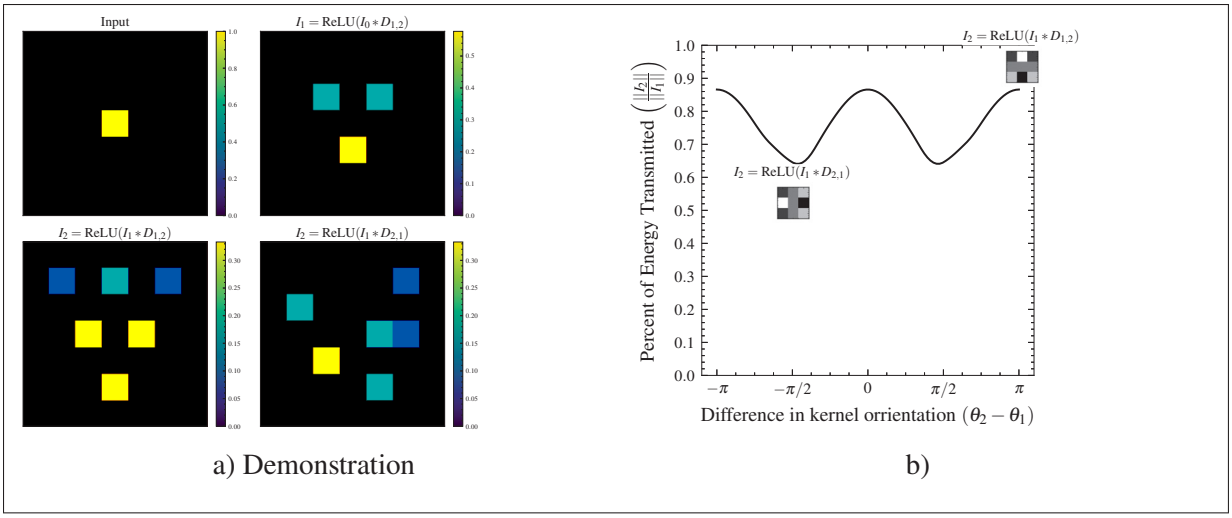
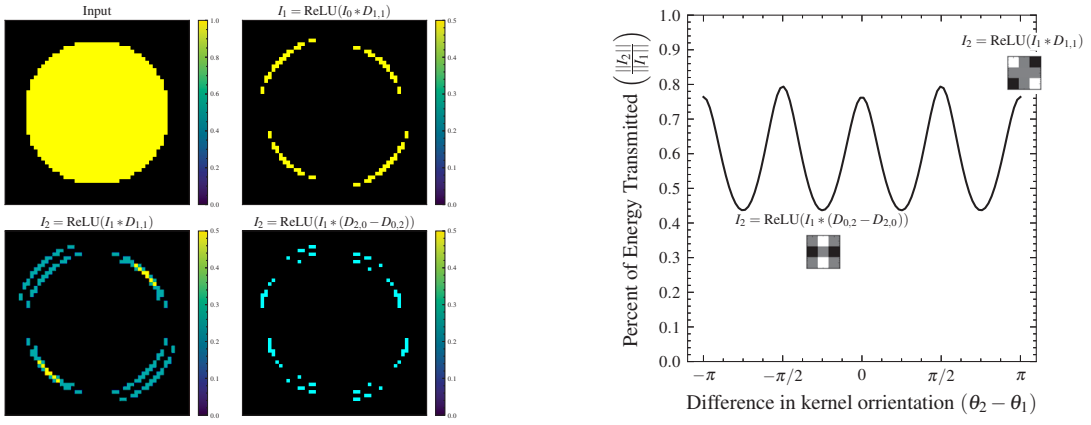


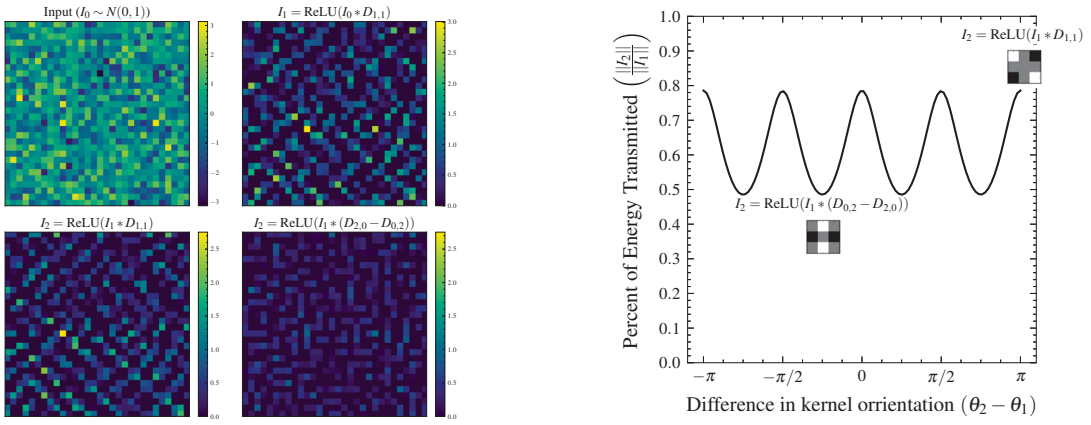
Figure-A I-20 (a) Sequential convolution with identical antisymmetric ($D_{1,2}$ and $D_{2,1}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a pixel test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = D_{1,2} \cos\theta + D_{2,1} \sin\theta$. Attenuation is maximal at an angular difference of $\pi/2$.

I.2.4 Sequential Convolution with Saddle $D_{1,1}$ at Varying Orientations



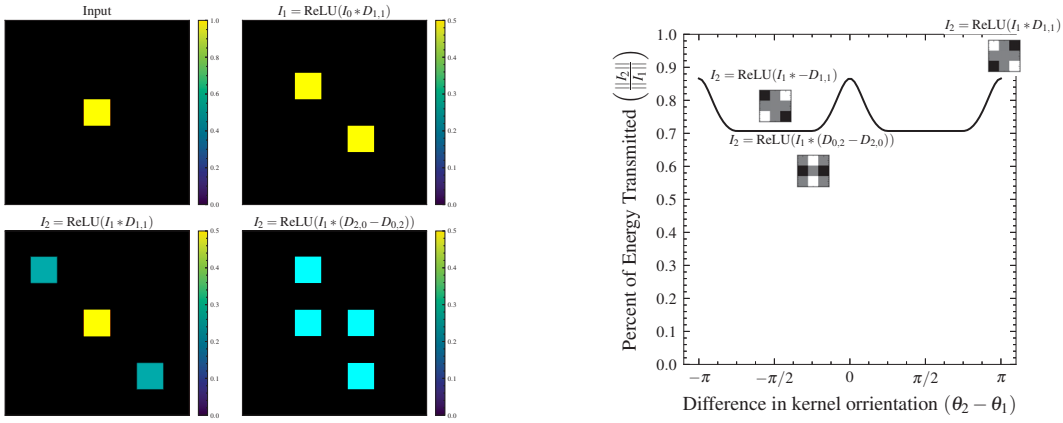
a) Demonstration b)

Figure-A I-21 (a) Sequential convolution with saddle/antisymmetric ($D_{1,1}$ and $D_{2,0} - D_{0,2}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a random test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = (D_{2,0} - D_{0,2}) \cos 2\theta + D_{1,1} \sin 2\theta$. Attenuation is maximal at an angular difference of $\pi/4$.



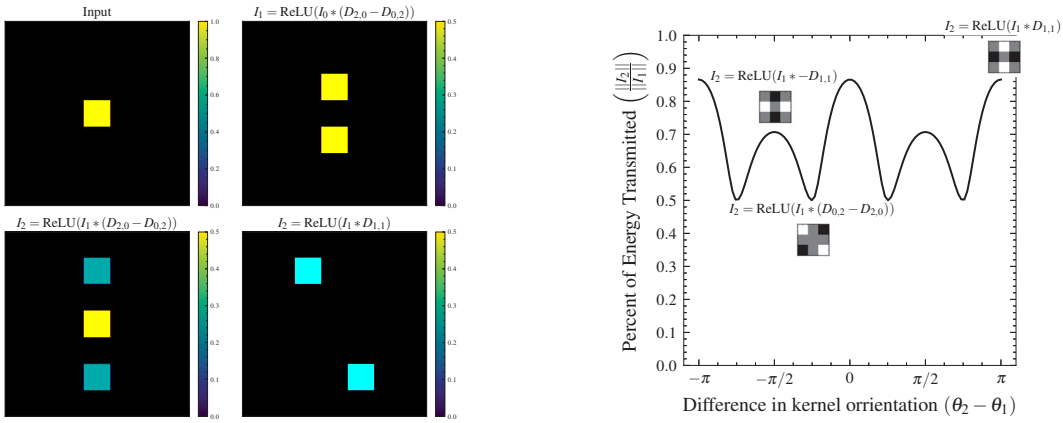
a) Demonstration b)

Figure-A I-22 (a) Sequential convolution with saddle/antisymmetric ($D_{1,1}$ and $D_{2,0} - D_{0,2}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a random test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = (D_{2,0} - D_{0,2}) \cos 2\theta + D_{1,1} \sin 2\theta$. Attenuation is maximal at an angular difference of $\pi/4$



a) Demonstration b)

Figure-A I-23 (a) Sequential convolution with saddle/antisymmetric ($D_{1,1}$ and $D_{2,0} - D_{0,2}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a circular test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = (D_{2,0} - D_{0,2}) \cos 2\theta + D_{1,1} \sin 2\theta$.



a) Demonstration b)

Figure-A I-24 (a) Sequential convolution with saddle/antisymmetric ($D_{1,1}$ and $D_{2,0} - D_{0,2}$) filters at varying relative orientations. Activation magnitude after applying two antisymmetric filters with angular difference $\theta_1 - \theta_2$, to a circular test pattern. Orthogonal filters ($\theta_1 - \theta_2 = \pi/2$) produce strong attenuation (lower right). (b) The second filter is steered as $F_\theta = (D_{2,0} - D_{0,2}) \sin 2\theta + D_{1,1} \cos 2\theta$.

APPENDIX II

CHAPTER 3 - ADDITIONAL RESULTS

In this appendix, we provide additional results from Chapter 3.

In Figures II-1 and II-2 we plot the average energy percentage for each frequency component ω_i of each kernel of VGG16 (Simonyan & Zisserman, 2015) and ResNet50 (He *et al.*, 2016), respectively, trained on ImageNet (Deng *et al.*, 2009) , across all layers.

In Figures II-3 and II-4 we plot the individual kernels of various filters, projected onto the antisymmetric plane. We observe that trained kernels have antisymmetric components that are correlated in terms of their orientations along a dominant orientation $\hat{\theta}$.

In Figures II-5 and II-6 we plot for various filter, their kernels projected onto the principal axis of the gradient (antisymmetric) plane ($\mathbf{v_1}$ (red)) and ($\mathbf{v_2}$ (blue)) along the horizontal axis and the symmetric energy along the vertical axis. We also plot the average (signed) symmetric energy in order to see the filters' DC distribution. We find that filters are usually mean-zero.

In Figure II-7 we plot the histograms of filter weights by channel (kernels in Layer L) to their respective filters (in Layer $L-1$), according to their gradient angular difference $(\theta_{F^L[i,:]} - \hat{\theta}_{F^{L-1}[:,i]})$. These histograms therefore quantify whether kernels of a given orientation θ tend to preferentially align with filters of similar orientation $\hat{\theta}$ in the preceding layer. We find that, in VGG16, there is a strong association between oriented filters and similarly oriented kernels across all layers.

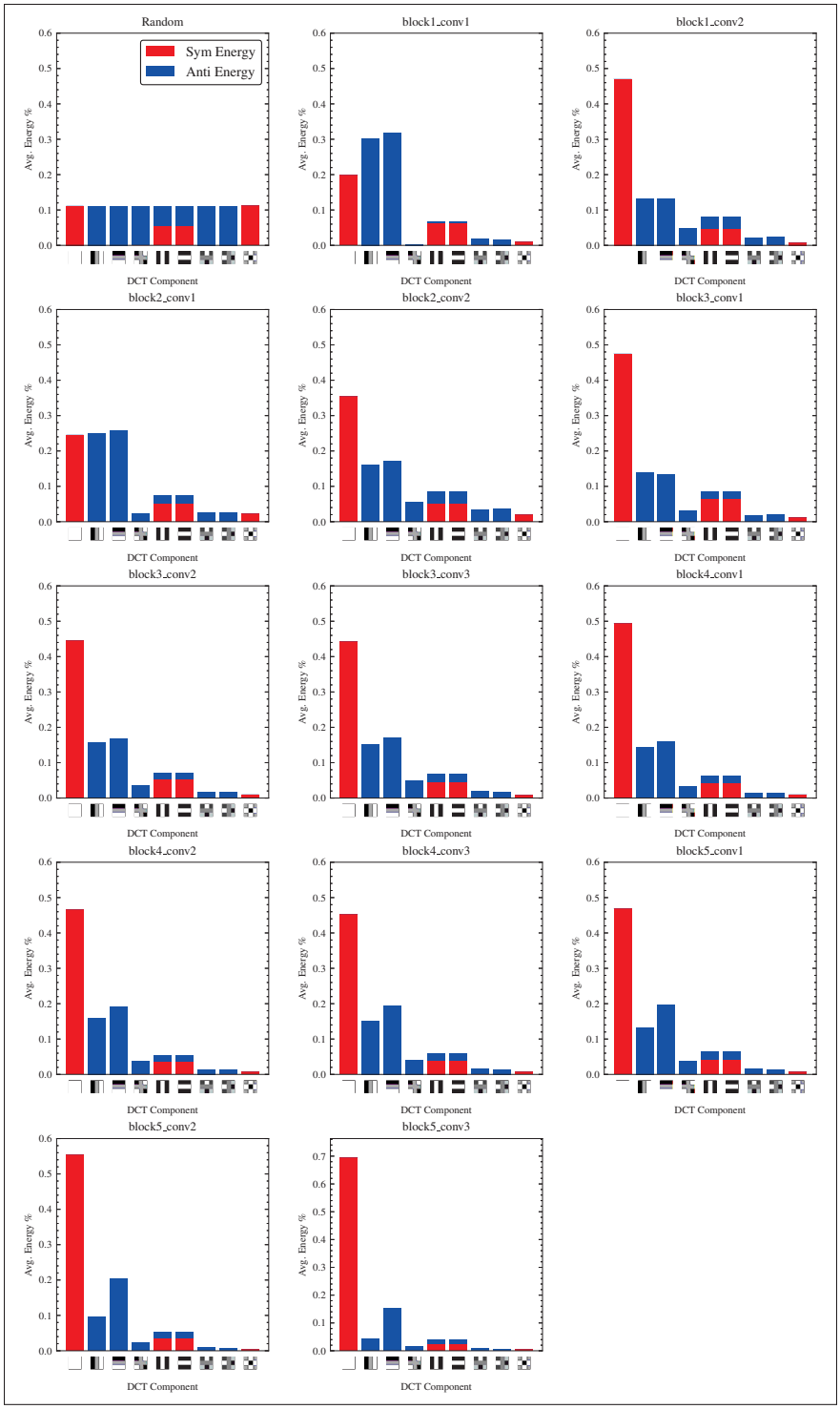


Figure-A II-1 Average energy distribution of DCT components ($\frac{\omega_i^2}{||\omega||^2}$) in random and learned convolutional kernels (trained on ImageNet) throughout VGG16

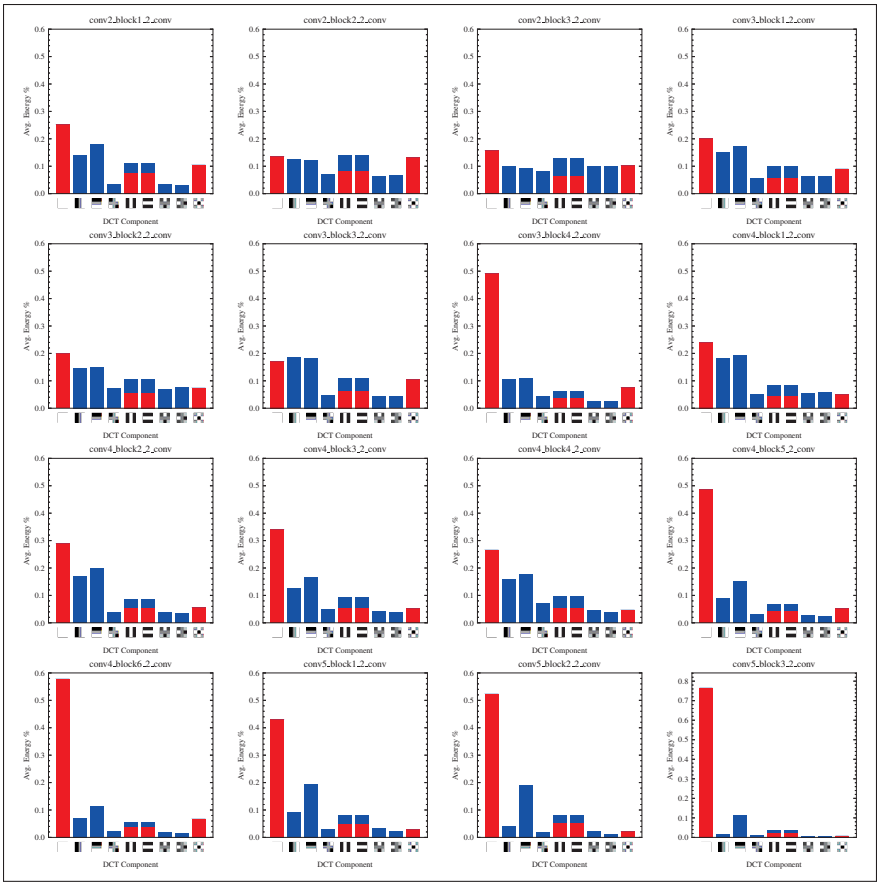


Figure-A II-2 Average energy distribution of DCT components ($\frac{\omega_i^2}{||\omega||^2}$) in learned convolutional kernels (trained on ImageNet) throughout ResNet50

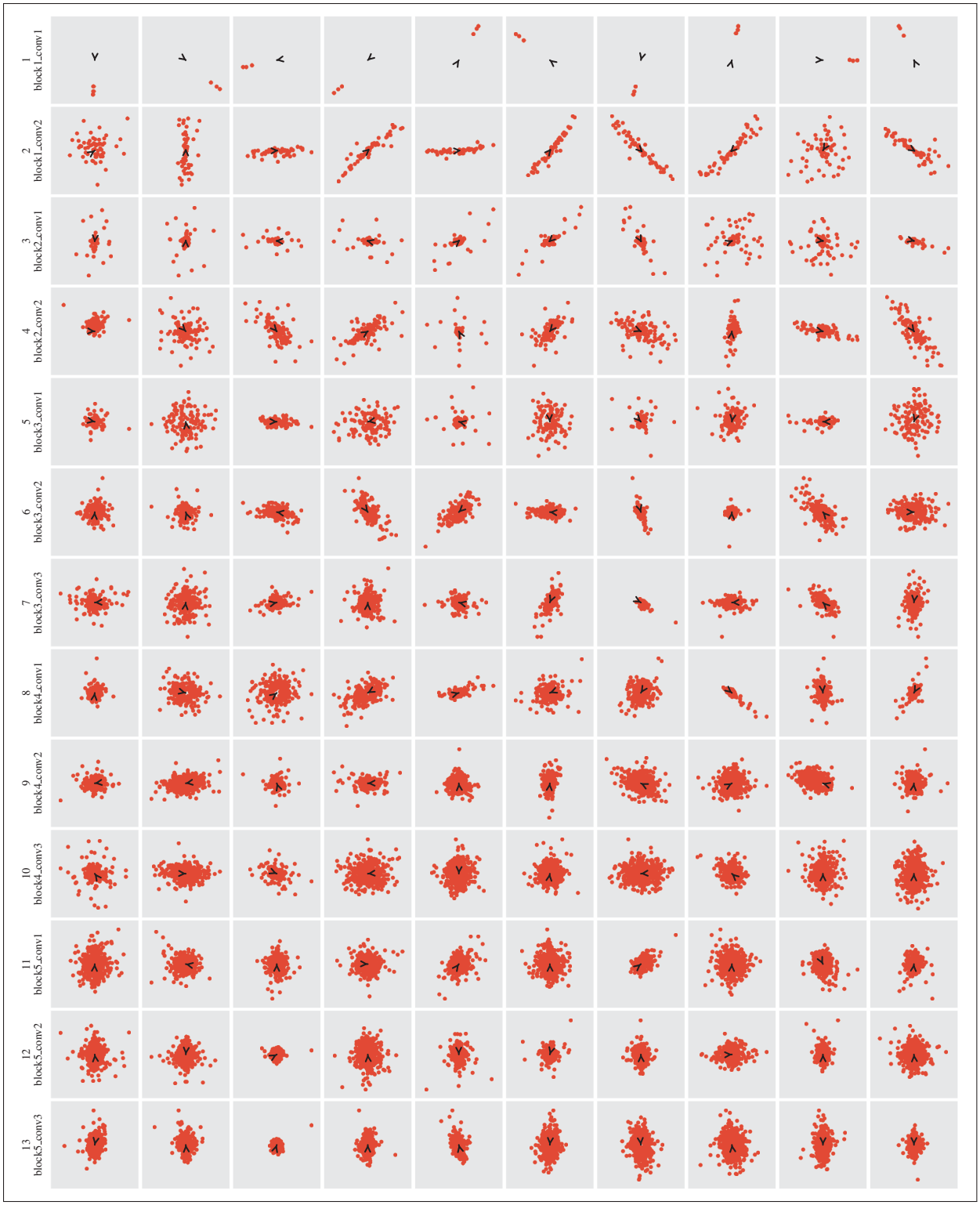


Figure-A II-3 Observing the antisymmetric plane of the top-10 VGG16 filters (ranked by ℓ^2-norm) trained on ImageNet (Deng *et al.*, 2009) across all the layers. Each point represents, in polar coordinates (θ, r_a), the orientation and antisymmetric magnitude of the kernels in a filter. The black arrow in the center denotes the pole, as well as the dominant orientation $\hat{\theta}$ of the filter.

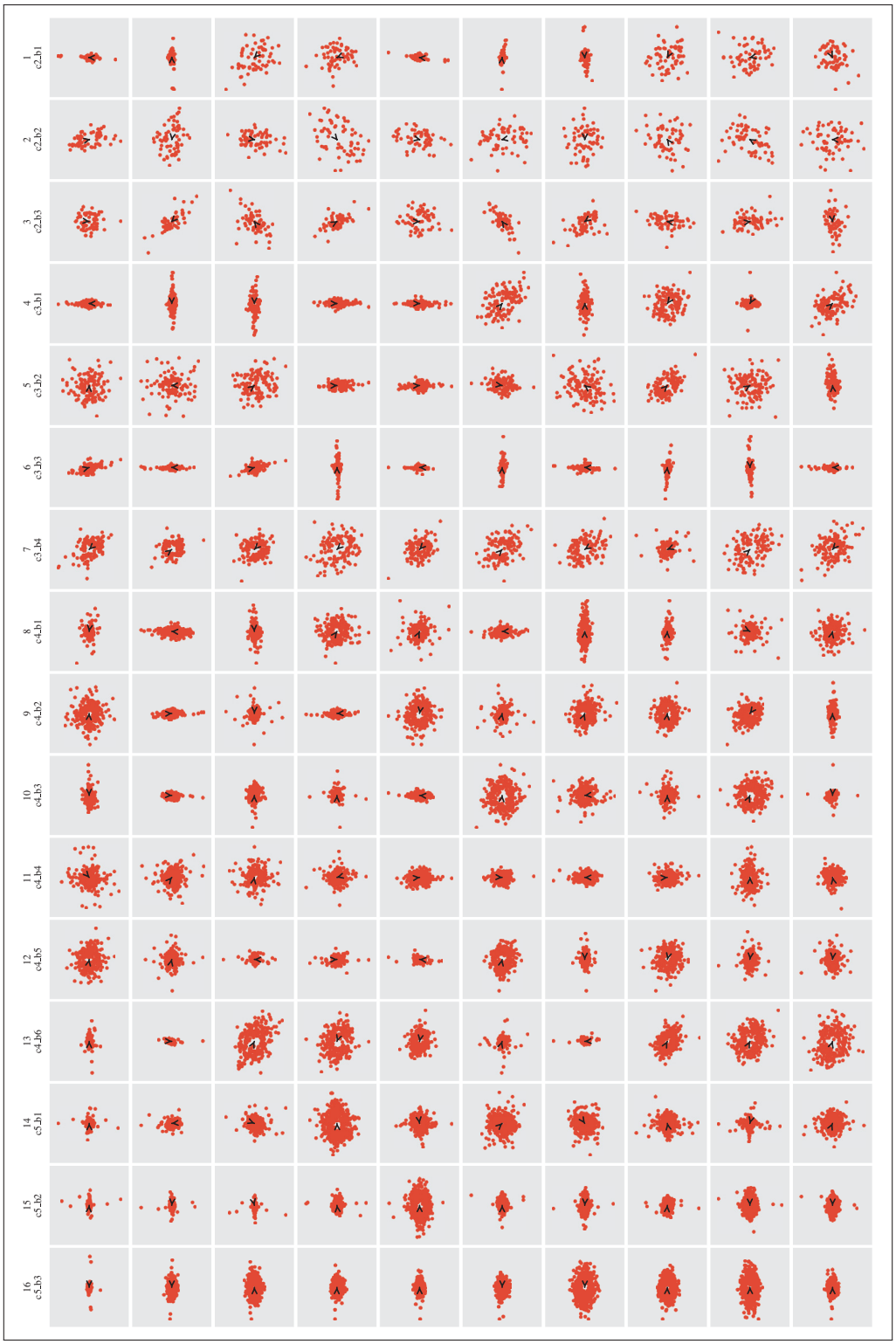


Figure-A II-4 Observing the antisymmetric plane of the top-10 ResNet filters (ranked by ℓ^2-norm) trained on ImageNet (Deng *et al.*, 2009) across all the layers. Each point represents, in polar coordinates (θ, r_a), the orientation and antisymmetric magnitude of the kernels in a filter. The black arrow in the center denotes the pole, as well as the dominant orientation $\hat{\theta}$ of the filter.

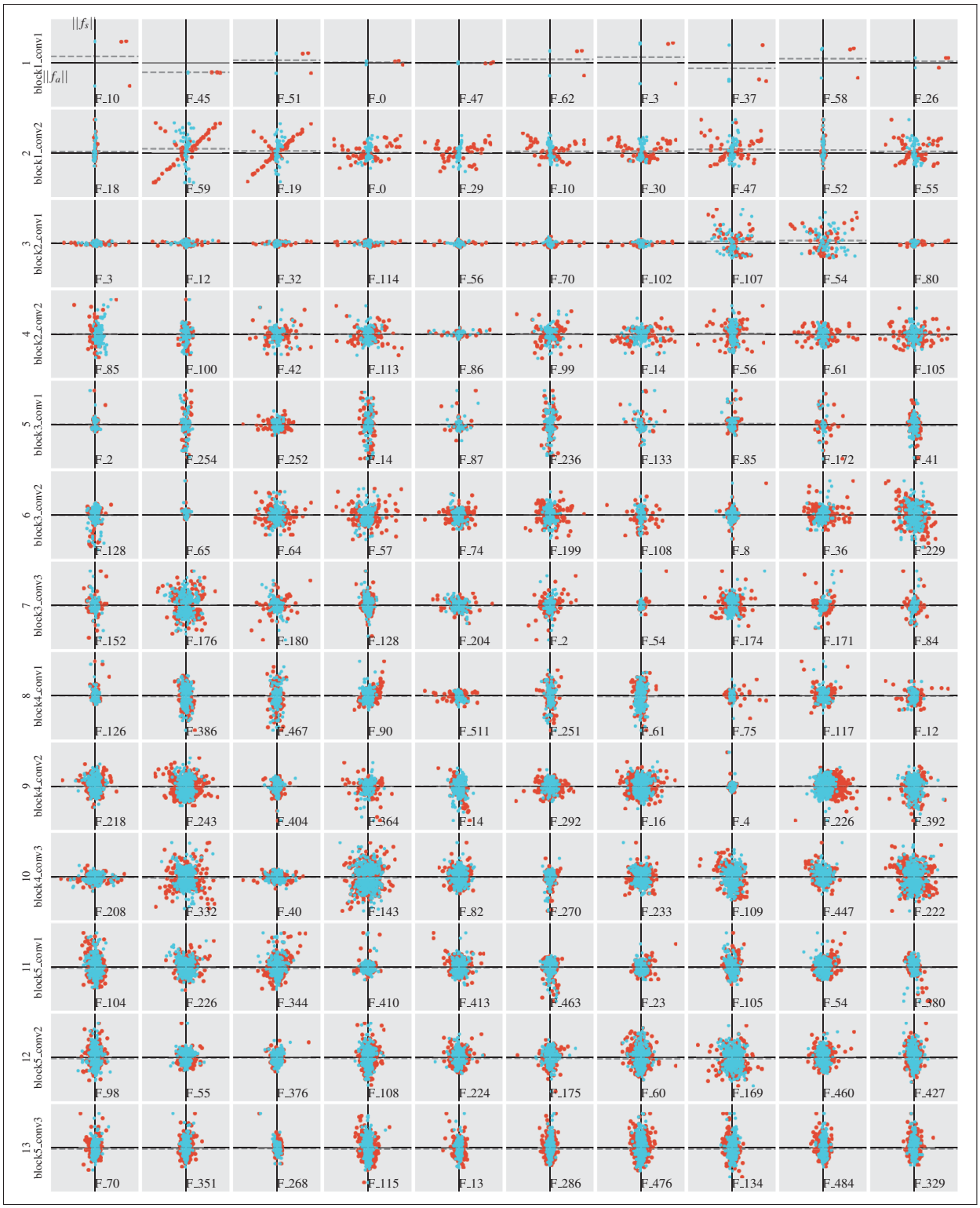


Figure-A II-5 VGG16 - Plotting kernels projected on to the principal axis of the gradient (antisymmetric) plane ($\mathbf{v_1}$ (red)) and ($\mathbf{v_2}$ (blue)) along the horizontal axis and the symmetric energy along the vertical axis. Observations are of the top-10 filters, ranked by ℓ^2-norm.

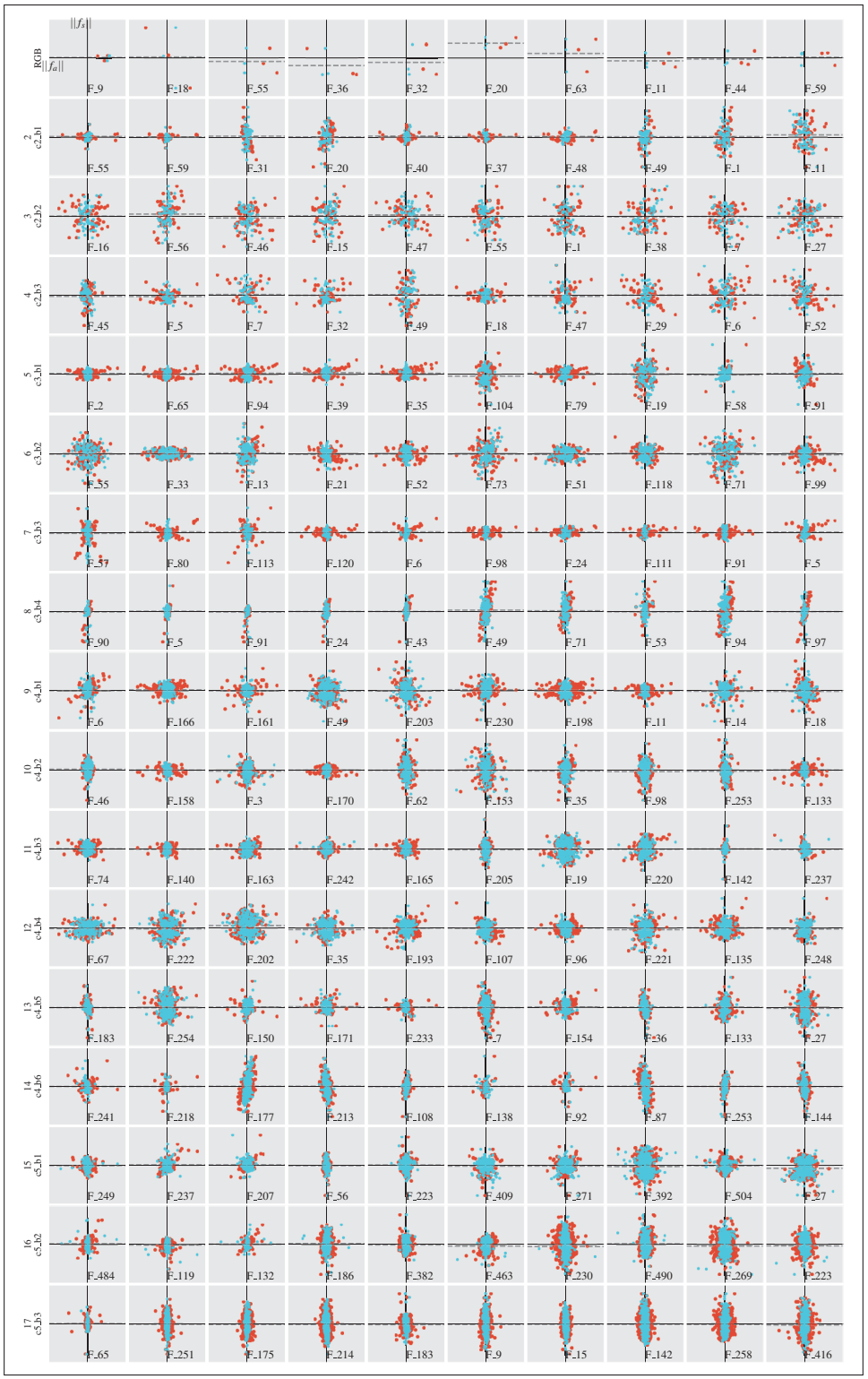


Figure-A II-6 ResNet50 - Plotting kernels projected on to the principal axis of the gradient (antisymmetric) plane ($\mathbf{v_1}$ (red)) and ($\mathbf{v_2}$ (blue)) along the horizontal axis and the symmetric energy along the vertical axis. Observations are of the top-10 filters, ranked by ℓ^2-norm.

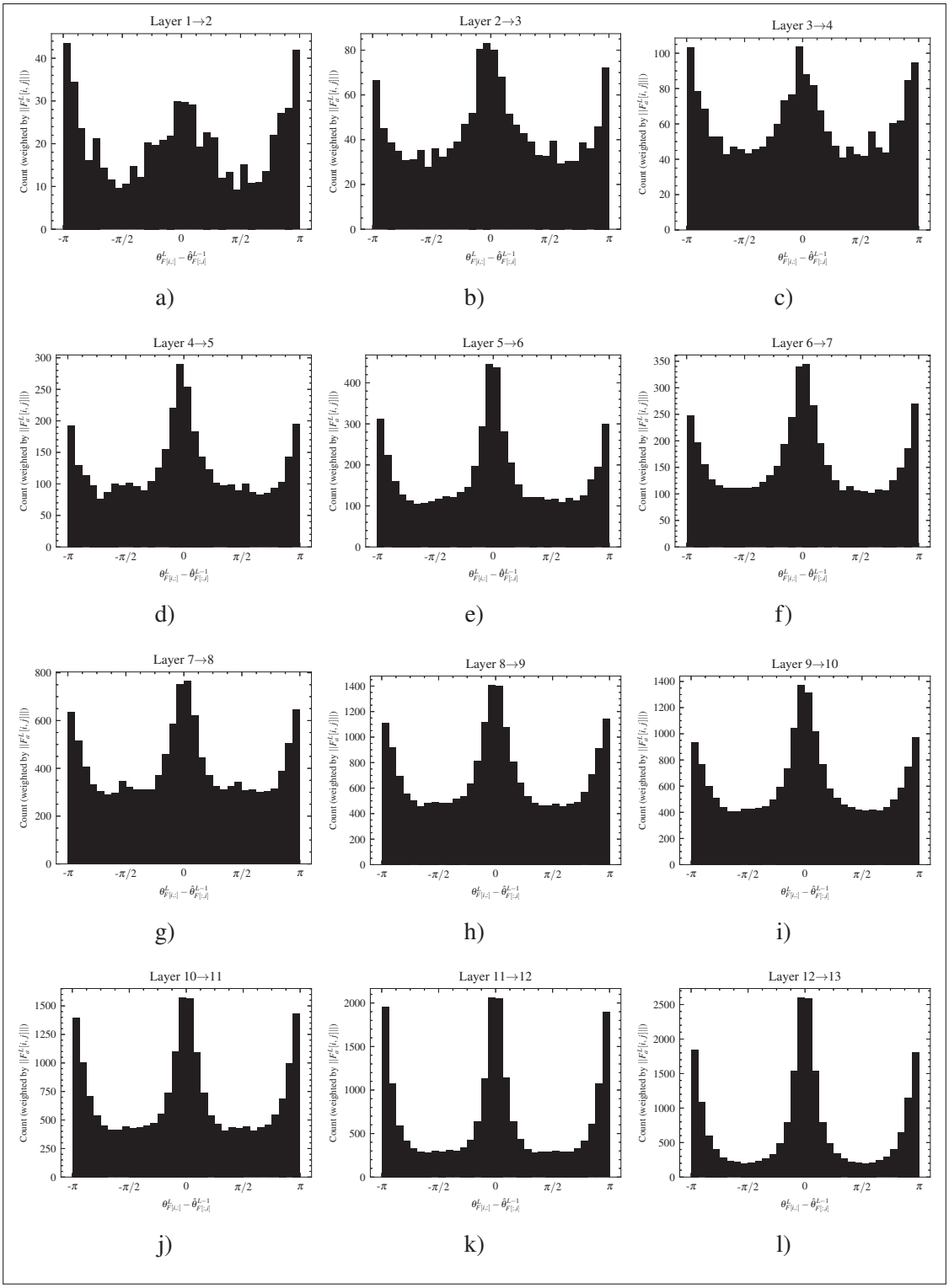


Figure-A II-7 Histograms of filter weights associated by channels (kernels in Layer L) to their respective filters (in Layer $L-1$) according to their gradient angular differences $(\theta_{F^L[i,:]} - \hat{\theta}_{F^{L-1}[:,i]})$ in VGG16

BIBLIOGRAPHY

Ahmed, N., Natarajan, T. and Rao, K. (1974). Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1), 90-93.

Alsallakh, B., Wroge, T., Miglani, V. and Kokhlikyan, N. (2025). On Symmetries in Convolutional Weights. *arXiv preprint arXiv:2503.19215*.

Aubury, M. and Luk, W. (1996). Binomial filters. *Journal of VLSI signal processing systems for signal, image and video technology*, 12(1), 35–50.

Bau, D., Zhu, J.Y., Strobelt, H., Lapedriza, A., Zhou, B. and Torralba, A. (2020). Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*.

Bell, A. and Sejnowski, T.J. (1996). Edges are the 'Independent Components'of Natural Scenes. *Advances in Neural Information Processing Systems*, 9.

Bereska, L. and Gavves, E. (2024). Mechanistic Interpretability for AI Safety – A Review. *arXiv e-prints*, arXiv:2404.14082.

Bruna, J. and Mallat, S. (2013). Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1872–1886.

Cammarata, N., Goh, G., Carter, S., Schubert, L., Petrov, M. and Olah, C. (2020). Curve Detectors. *Distill*.

Cammarata, N., Goh, G., Carter, S., Voss, C., Schubert, L. and Olah, C. (2021). Curve Circuits. *Distill*.

Chęiński, K. and Wawrzyński, P. (2020). DCT-Conv: Coding filters in convolutional networks with Discrete Cosine Transform. *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6.

Chollet, F. et al. (2015). Keras.

Chowers, R. and Weiss, Y. (2023). What do CNNs learn in the first layer and why? a linear systems perspective. *International Conference on Machine Learning*, pp. 6115–6139.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

Erhan, D., Bengio, Y., Courville, A. and Vincent, P. (2009). Visualizing Higher-Layer Features of a Deep Network. *Technical Report, Univeristé de Montréal*.

Freeman, W.T., Adelson, E.H. et al. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9), 891–906.

Fu, L. (1991). Rule Learning by Searching on Adapted Nets. *AAAI Conference on Artificial Intelligence*.

Fukushima, K. (2013). Artificial vision by multi-layered neural networks: Neocognitron and its advances. *Neural networks*, 37, 103–119.

Fukuzaki, S. and Ikehara, M. (2022). Principal Components of Neural Convolution Filters. *IEEE Access*, 10, 104328-104336.

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26), 429–441.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Glorot, X., Bordes, A. and Bengio, Y. (2011). Deep sparse rectifier neural networks. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.

Haar, A. (1909). *Zur theorie der orthogonalen funktionensysteme*. Georg-August-Universitat, Gottingen.

He, K., Zhang, X., Ren, S. and Sun, J. (2015, December). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hinton, G.E., McClelland, J.L. and Rumelhart, D.E. (1986). Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations* (pp. 77–109). Cambridge, MA, USA: MIT Press.

Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160, 106–154.

Kechris, C., Dan, J., Miranda, J. and Atienza, D. (2024). Dc is all you need: describing relu from a signal processing standpoint. *arXiv preprint arXiv:2407.16556*.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541-551.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.

Li, H., Kadav, A., Durdanovic, I., Samet, H. and Graf, H.P. (2017). Pruning Filters for Efficient ConvNets. *International Conference on Learning Representations*.

Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2), 225–270.

Lowe, D.G. (1999). Object recognition from local scale-invariant features. *Proceedings of the seventh IEEE international conference on computer vision*, 2, 1150–1157.

Luo, W., Li, Y., Urtasun, R. and Zemel, R. (2016). Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 29.

Maas, A.L., Hannun, A.Y., Ng, A.Y. et al. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proc. icml*, 30(1), 3.

Minsky, M.L. and Papert, S.A. (1988). *Perceptrons: expanded edition*. Cambridge, MA, USA: MIT Press.

Nair, V. and Hinton, G.E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K. and Mordvintsev, A. (2018). The Building Blocks of Interpretability. *Distill*.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M. and Carter, S. (2020a). An Overview of Early Vision in InceptionV1. *Distill*.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M. and Carter, S. (2020b). Zoom in: An introduction to circuits. *Distill*, 5(3), e00024–001.

Olah, C., Cammarata, N., Voss, C., Schubert, L. and Goh, G. (2020c). Naturally Occurring Equivariance in Neural Networks. *Distill*.

Petrov, M., Voss, C., Schubert, L., Cammarata, N., Goh, G. and Olah, C. (2021). Weight Banding. *Distill*.

Rippel, O., Snoek, J. and Adams, R.P. (2015). Spectral Representations for Convolutional Neural Networks.

Ruderman, D. and Bialek, W. (1993). Statistics of Natural Images: Scaling in the Woods. *Advances in Neural Information Processing Systems*, 6.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5), 206–215.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1985). *Learning internal representations by error propagation*.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations*.

Simonyan, K., Vedaldi, A. and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning*, pp. 2256–2265.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u. and Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30.

Voss, C., Cammarata, N., Goh, G., Petrov, M., Schubert, L., Egan, B., Lim, S.K. and Olah, C. (2021). Visualizing Weights. *Distill*.

Wallace, G.K. (1991). The JPEG still picture compression standard. *Commun. ACM*, 34(4), 30–44.

Weiler, M., Hamprecht, F.A. and Storath, M. (2018). Learning Steerable Filters for Rotation Equivariant CNNs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 849-858.

Wu, S., Wang, G., Tang, P., Chen, F. and Shi, L. (2019). Convolution with even-sized kernels and symmetric padding. *Advances in Neural Information Processing Systems*, 32.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). How 'transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Zeiler, M.D. and Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *Computer Vision – ECCV 2014*, pp. 818–833.

Zhang, Y., Tiňo, P., Leonardis, A. and Tang, K. (2021). A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5), 726-742.