PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure de Paris

Towards a Mathematical Understanding of Deep Convolutional Neural Networks

Soutenue par
Florentin Guth
Le 29 Août 2023

École doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité
Mathématiques appliquées

Composition du jury :

Lorenzo Rosasco Professor, Univ. Genova	*Rapporteur*
Eric Vanden-Eijnden Professor, Courant Institute, NYU	*Rapporteur*
Francis Bach Professeur, INRIA	*Président du jury*
Giulio Biroli Professeur, ENS	*Examinateur*
Marylou Gabrié Assistant Professor, Polytechnique	*Examinatrice*
Stéphane Mallat Professeur, Collège de France	*Directeur de thèse*

1794
ENS | PSL

À mes parents, Marie-Françoise et Claude.

Résumé

Les réseaux de neurones convolutifs profonds ont obtenu un succès considérable en vision par ordinateur, à la fois pour l'apprentissage non-supervisé (i.e., génération d'image) et l'apprentissage supervisé (i.e., classification d'image). Cependant, les principes fondamentaux derrière ces résultats impressionnants ne sont pas bien compris. En particulier, l'apprentissage profond semble échapper à la malédiction de la dimensionalité, ce qui révèle une structure mathématique riche dans les problèmes d'apprentissage rencontrés en pratique. Cette structure est présente dans les interactions entre les données d'entraînement (sur quelles propriétés se repose-t-on implicitement ?), l'architecture (quel est le rôle fonctionnel rempli par ses composants ?) et l'algorithme d'optimisation (qu'est-ce que le réseau a appris ?). Cette thèse comporte des résultats sur ces trois questions. Premièrement, nous montrons qu'une factorisation multi-échelles des distributions d'images peut révéler des propriétés de régularité, des structures de dépendances markoviennes locales, et même de la log-concavité conditionnelle, alors que la distribution globale ne possède pas ces propriétés. Cela conduit à des algorithmes efficaces d'apprentissage et d'échantillonnage dont on peut contrôler toutes les sources d'erreurs. Deuxièmement, nous étudions le rôle de la non-linéarité en classification d'images, et montrons que sa fonction principale est de collapser la phase complexe des coefficients d'ondelettes des activations du réseau. En revanche, des modèles précédents reposant sur des seuillages et des hypothèses de parcimonie ne sont ni suffisants ni nécessaires pour expliquer la précision de classification des réseaux profonds. Troisièmement, nous introduisons un modèle probabiliste des poids appris dans les architecture profondes, en capturant les dépendances entre couches par un alignement des activations du réseau sur une représentation déterministe associée à un noyau reproduisant. Le modèle est spécifié à travers des distributions à chaque couche, dont les covariances sont de bas rang et réalisent une réduction de dimensionalité entre les plongements en haute dimension calculés par la non-linéarité. Dans certains cas, ces distributions sont approximativement gaussiennes, et leurs covariances capturent la performance et la dynamique d'entraînement du réseau.

réseaux de neurones convolutifs ⋆ apprentissage profond ⋆ vision par ordinateur ⋆ classification d'images ⋆ génération d'images ⋆ représentations multi-échelles

Abstract

Deep convolutional neural networks have achieved considerable success in computer vision tasks, both in unsupervised learning (e.g., image generation) and supervised learning (e.g., image classification). However, the fundamental principles behind these impressive results remain not well understood. In particular, deep learning seemingly escapes the curse of dimensionality in practice, which evidences a rich mathematical structure underlying real-world learning problems. This structure is revealed by the interplay between the training data (what properties are we implicitly relying on?), the architecture (what is the functional role of network computations?), and the optimization algorithm (what has the network learned?). This thesis presents results on these three questions. First, we demonstrate that a multiscale factorization of image distributions can reveal properties of smoothness, local Markov dependency structure, and even conditional log-concavity, whereas the global distribution does not enjoy these properties. It leads to efficient learning and sampling algorithms where all sources of errors can be controlled. Second, we investigate the role of non-linearity in image classification, and show that its main function is to collapse the phase of complex wavelet coefficients of network activations. In contrast, previous models based on thresholding and sparsity assumptions are neither sufficient nor necessary to explain the classification accuracy of deep networks. Third, we introduce a probabilistic model of learned weights in deep architectures, with layer dependencies that are captured by alignment of the network activations to deterministic kernel embeddings. The model is specified through weight distributions at each layer, whose covariances are low-rank and perform dimensionality reduction in-between the high-dimensional embeddings computed by the non-linearities. In some cases, these weight distributions are approximately Gaussian, and their covariances capture the performance and training dynamics of the network.

convolutional neural networks ⋆ deep learning ⋆ computer vision ⋆ image classification ⋆ image generative modeling ⋆ multiscale representations

Remerciements

J'aimerais commencer ce manuscrit en remerciant Stéphane. Stéphane, merci profondément pour ces quatre ans et demi de thèse. J'ai tant appris à tes côtés, graĉe à ta disponibilité, ton investissemment et ton exigence. En particulier, tu m'as transmis ta passion de la recherche, et je t'en suis extrêmement reconnaissant. Merci pour toutes les discussions au tableau dans ton bureau dont je suis sorti avec des étoiles dans les yeux. Je continuerai longtemps d'être inspiré par ton talent, ton énergie et ta vision scientifique. Merci infiniment.

Je voudrais aussi remercier Brice Ménard, qui a été comme un second directeur de thèse pour moi. Brice, tu m'as enseigné à raisonner et voir le monde comme un physicien, et ce que j'ai appris grâce à toi va bien plus loin que la recherche. Merci pour nos nombreuses discussions à toute heure du jour et de la nuit, où n'importe quel sujet est à portée d'une connection ou analogie. Merci aussi pour tes conseils, tes encouragements et ton soutien. Cette thèse ne serait pas la même si tu n'était pas venu passer un an à l'ENS. Je te dois beaucoup.

Plus largement, j'ai bénéficié durant ma thèse des conseils de plusieurs personnes dont je suis reconaissant. Je remercie Francis Bach de m'avoir suivi depuis ma scolarité à l'ENS, et pour m'avoir maintes fois aidé dans mes démarches ou ma recherche. Tomás, merci pour tes nombreuses explications patientes pendant mon stage dans l'équipe, tu as été un mentor pour moi. And finally, thank you Eero for your precious career advice.

Je voudrais aussi remercier les professeurs qui m'ont donnée envie de poursuivre une carrière scientifique. Merci à M. Miguet, M. Buffenoir et Mme Arnaud. Merci en particulier à mes professeurs de maths du Lycée du Parc, Franz Ridde et Denis Choimet, pour m'avoir donné le goût d'apprendre et de comprendre, et d'avoir été des professeurs exceptionnels.

I would like to warmly thank Lorenzo Rosasco and Eric Vanden-Eijnden for agreeing to write a report on this manuscript—without knowing what they were getting into! Merci également aux membres du jury d'avoir accepté d'en faire partie : Francis Bach, Giulio Biroli, Marylou Gabrié.

Je salue tous les membres de la joyeuse équipe DATA à l'ENS, qui ont grandement égayé mon séjour dans le labo. Merci aux "grands", Edouard, Alberto, Roberto, Tanguy et Simon, et aux "petits", Samuel, Etienne et Nathanaël. Merci à Tomás pour les discussions philosophiques, à Louis pour son sens de la formule inégalé, et à Antoine pour les cours d'informatique. Merci particulièrement à John pour les débats politiques et déontologiques. Enfin, je remercie mes acolytes de thèse : Rudy, pour relever le niveau d'humour du labo, et Gaspar, pour le rabaisser. J'ai eu de la chance de faire ma thèse en même temps que vous.

Plus largement, je souhaite aussi remercier tous les membres du CSD qui ont contribué à en faire un lieu de travail agréable. Merci à Zaccharie pour sa co-organisation des séminaires d'une efficacité redoutable, c'était un plaisir de faire ça avec toi ! Merci à Rudy mon co-fondateur du random lunch pour avoir été là pendant toutes nos péripéties administratives, et merci à Pablo et Etienne d'avoir repris le flambeau. Merci à Gabriel et Bruno pour leur disponibilité et leurs conseils experts. Et pour l'ambiance toujours au top, merci à Maria, Léa, Pablo, Mathieu, Michael, Othmane, Raphaël.

I have been lucky to spend an awesome summer at the Flatiron Institute. Stéphane, thanks a lot for this amazing opportunity. I would like to thank Eero and Matthew, who truly went out of their way to make me feel welcome. Thanks to everyone at CCN who made my stay a great

time: Zahra for the both fascinating and fun collaboration, Pierre-Étienne pour ton amitié, Ben for the many late-night discussions (except when it was about desacrating French gastronomy), and for contributing to the lively atmosphere: Lyndon, Jules, David, Siavash, Teddy, Nikhil. I also thank Bargeen, Mashail, Melissa for the fun times we had together.

This summer in the US was also the occasion to visit Johns Hopkins: thank you Brice for the invitation and your hospitality. I would like to thank the Bonner Lab for the many inspiring discussions. In particular, thank you Mick for your very generous offer to share a keynote and tutorial at the CCN conference, and thanks a lot for your valuable feedback and help. This was an intense sprint, with my defense being three days later not exactly helping, but this was extremely rewarding! I also extend special thanks to the amazing team for their tireless efforts in bringing our tutorial together: Raj, Atlas, Ray. This was a lot of fun thanks to you!

Je voudrais également remercier mes co-auteurs pour leur intensité à toute épreuve dans les deadlines, parfois jusqu'à la nuit blanche : merci à John, Zahra, Etienne, Tomás, Eero, Simon, Valentin.

J'ai eu de nombreuses opportunités d'enseigner pendant ma thèse, et je voudrais remercier mes élèves pour leur motivation et leur participation. Merci aux élèves du CPES d'être particulièrement adorables, et à l'équipe de profs et chargés de TDs pour avoir toujours été disponible pour des remplacements : Sébastien, Cyrille, Pierre, Thibault, Pierre, Thomas, Raphaël, Etienne, Yafei. Merci aussi à Antoine Lamy et Bertrand Léonard de m'avoir accueuilli à Optimal Sup-Spé puis Ipesup, et de m'avoir fait confiance pour créer un stage d'initiation à l'IA pour lycéens à partir de rien. Enfin, merci à Lolo pour son sponsoring.

Je n'ai pas appris qu'à faire de la recherche durant cette thèse. Je voudrais remercier mes professeurs de musique pour tout ce qu'ils m'ont enseigné : Claire, Hugues, Il-Woong. En particulier, je tiens à remercier chaleureusement Claire pour sa capacité à toujours trouver les bons mots et sa gentillesse. Les cours et la pratique des instruments m'ont permis de faire des pauses, recharger les batteries dans les moments difficiles, et repartir avec de la musique plein la tête : je leur suis très reconnaissant pour cela.

Cette thèse doit également beaucoup à diverses sources de réconfort matériel qui ne m'ont jamais abandonné : merci aux boulangeries du quartier, aux japonais à volonté, et aux restaus du coin dont j'ai fini par enregistrer le numéro dans mes contacts. Merci aussi aux sitcoms et à la team du lundi pour le divertissement.

Merci à ma famille de créer un climat si agréable où l'on se sent apprécié quoi qu'il arrive, et pour tous les bons moments passés à Frangy avec les cousins.

J'adresse mes remerciements aux amis de l'ENS, pour leur compagnie toujours très agréable : merci à Suzanne, Lucas, Pauline, Jules. Merci aussi aux info16 pour les retrouvailles régulières : Marc, Matthieu, Luc, Lucas P, Lucas W. Je remercie également les compères Martin et Rudy pour les Nouvel An improvisés et moins improvisés, les soirées jeux et les sorties parisiennes. Enfin, merci aux zigotos du Bolmen Palace : Jean-Paul Robin, pour le role-play grincheux, pour les lamentations chroniques, les parodies musicales pourries, et les imitations baveuses; et Lionel Zoubritzky, pour le role-play chose, pour l'art du malaise, pour m'aider à faire tourner Paul en bourrique, et pour les retournements de veste inévitables. Merci à tous les deux pour nos partenariats commerciaux avec Jojo et Lolo, pour les discussions d'un intérêt douteux, le Pâtes Thaï fermé le dimanche soir, et pour Potiche McPotFace.

Pour finir, je voudrais remercier Roro, Papa et Maman pour leur soutien indéfectible.

Table of contents

Résumé iii

Abstract v

Remerciements vii

1 Introduction 1
- 1.1 Curse of dimensionality and structure in computer vision . . . 2
  - 1.1.1 Learning models of high-dimensional probability distributions . . . 2
  - 1.1.2 The curse of dimensionality in supervised learning . . . 4
  - 1.1.3 The curse of dimensionality in unsupervised learning . . . 5
  - 1.1.4 Deep convolutional neural networks . . . 5
  - 1.1.5 Leveraging structure to escape the curse of dimensionality . . . 7
- 1.2 Properties of wavelet conditional probability distributions . . . 10
  - 1.2.1 Score-based diffusions and autoregressive factorizations . . . 10
  - 1.2.2 Conditional log-concavity of physical fields . . . 13
  - 1.2.3 Conditional locality and regularity of natural images . . . 15
- 1.3 Non-linear operators for image classification . . . 16
  - 1.3.1 Separation and concentration in deep networks . . . 17
  - 1.3.2 Concentration with thresholdings in sparse representations . . . 18
  - 1.3.3 Separation with phase collapses of wavelet coefficients . . . 20
- 1.4 A model of network weights with aligned random features . . . 22
  - 1.4.1 Random-feature kernels in deep networks . . . 22
  - 1.4.2 Evolution of kernels and training dynamics . . . 23
  - 1.4.3 Alignment convergence: the rainbow model . . . 24
- 1.5 Organization of the dissertation . . . 27

I Properties of Wavelet Conditional Probability Distributions 29

2 Conditionally Strongly Log-Concave Generative Models 31
- 2.1 Introduction . . . 32
- 2.2 Conditionally strongly log-concave models . . . 33
  - 2.2.1 Conditional factorization and log-concavity . . . 33
  - 2.2.2 Learning guarantees with score matching . . . 35
  - 2.2.3 Score matching with exponential families . . . 36
  - 2.2.4 Sampling guarantees with MALA . . . 37
- 2.3 Wavelet packet conditional log-concavity . . . 38
  - 2.3.1 Energies with scalar potentials . . . 38
  - 2.3.2 Wavelet packets and renormalization group . . . 39
  - 2.3.3 Multiscale scalar potentials . . . 39
- 2.4 Numerical results . . . 40

2.4.1 φ^4 scalar potential energy . . . 40
2.4.2 Conditional log-concavity . . . 41
2.4.3 Application to cosmological data . . . 43
2.5 Discussion . . . 44

3 Wavelet Score-Based Generative Models 45
3.1 Introduction . . . 46
3.2 Sampling and discretization of score-based generative models . . . 47
3.2.1 Score-based generative models . . . 47
3.2.2 Discretization of SGMs and score regularity . . . 48
3.3 Wavelet score-based generative models . . . 50
3.3.1 Wavelet whitening and cascaded SGMs . . . 50
3.3.2 Discretization and accuracy for Gaussian processes . . . 52
3.4 Acceleration with WSGM: numerical results . . . 53
3.4.1 Physical processes with scalar potentials . . . 53
3.4.2 Scale-wise time reduction in natural images . . . 54
3.5 Discussion . . . 56

4 Multiscale Local Conditional Models of Images 57
4.1 Introduction . . . 57
4.2 Markov wavelet conditional models . . . 59
4.3 Score-based markov wavelet conditional models . . . 60
4.4 Markov wavelet conditional denoising . . . 62
4.5 Markov wavelet conditional super-resolution and synthesis . . . 64
4.6 Discussion . . . 66

II Non-Linear Operators for Image Classification 69

5 Separation and Concentration in Deep Networks 71
5.1 Introduction . . . 72
5.2 Classification by separation and concentration . . . 72
5.2.1 Tight frame rectification and thresholding . . . 72
5.2.2 Two-layer networks without bias . . . 75
5.3 Deep learning by scattering and concentrating . . . 77
5.3.1 Scattering cascade of wavelet frame separations . . . 77
5.3.2 Separation and concentration in learned scattering networks . . . 79
5.4 Discussion . . . 81

6 Phase Collapse in Deep Networks 83
6.1 Introduction . . . 83
6.2 Eliminating spatial variability with phase collapses . . . 85
6.3 Learned scattering network with phase collapses . . . 86
6.4 Phase collapses versus amplitude reductions . . . 88
6.5 Iterating phase collapses and amplitude reductions . . . 91
6.5.1 Iterated phase collapses . . . 91
6.5.2 Iterated amplitude reductions . . . 92
6.6 Discussion . . . 93

III A Model of Network Weights with Aligned Random Features **95**

7 The Rainbow Model of Deep Networks **97**
7.1 Introduction . . . 98
7.2 Rainbow networks . . . 100
7.2.1 Rotations in random feature maps . . . 100
7.2.2 Deep rainbow networks . . . 103
7.2.3 Symmetries and convolutional rainbow networks . . . 109
7.3 Numerical results . . . 111
7.3.1 Convergence of activations in the infinite-width limit . . . 112
7.3.2 Properties of learned weight covariances . . . 114
7.3.3 Gaussian rainbow approximations . . . 120
7.4 Discussion . . . 125

Conclusion **129**

8 Conclusion **129**
8.1 Summary of findings . . . 129
8.2 Perspectives . . . 130

Appendices **135**

A Appendix for Chapter 2 **135**
A.1 Definition of wavelet packet projectors . . . 135
A.1.1 Conjugate mirror filters . . . 135
A.1.2 Orthogonal frequency decomposition . . . 136
A.1.3 Wavelet packet projectors . . . 137
A.2 Score matching and MALA algorithms for CSLC exponential families . . . 137
A.2.1 Multiscale energies . . . 137
A.2.2 Pseudocode . . . 140
A.3 Experimental details . . . 140
A.3.1 Datasets . . . 140
A.3.2 Experimental setup . . . 141
A.3.3 Mixing times in MALA . . . 141
A.4 Energy estimation with free-energy modeling . . . 142
A.4.1 Free-energy score matching . . . 142
A.4.2 Parameterized free-energy models . . . 142
A.4.3 Multiscale energy decomposition . . . 143
A.5 Proof of Proposition 2.3 . . . 144

B Appendix for Chapter 3 **145**
B.1 WSGM algorithm . . . 145
B.2 Introduction to the fast orthogonal wavelet transform . . . 146
B.3 Experimental details on Gaussian experiments . . . 147
B.4 Experimental details on the φ^4 model . . . 148
B.5 Experimental details on CelebA-HQ . . . 148

C Appendix for Chapter 4 **151**
C.1 Proof of Theorem 4.1 . . . 151
C.2 Proof of equation (4.5) . . . 152
C.3 Training and architecture details . . . 152
C.4 Wavelet conditional synthesis algorithm . . . 153

D Appendix for Chapter 5 **155**
D.1 Proof of Proposition 5.1 . . . 155
D.2 Proof of Theorem 5.2 . . . 156
D.3 Implementation and network dimensions . . . 156

E Appendix for Chapter 6 **159**
E.1 Proof of Theorem 6.1 . . . 159
E.2 Proof of equation (6.4) . . . 159
E.3 Proof of Theorem 6.2 . . . 160
E.4 Proof of Theorem 6.3 . . . 161
E.5 Experimental details . . . 162

F Appendix for Chapter 7 **165**
F.1 Proof of Theorem 7.1 . . . 165
F.1.1 Proof outline . . . 165
F.1.2 Proof of Lemma F.1 . . . 168
F.1.3 Proof of Lemma F.2 . . . 169
F.1.4 Proof of Lemma F.3 . . . 169
F.1.5 Proof of Lemma F.4 . . . 171
F.2 Proof of Theorem 7.2 . . . 171
F.3 Proof of Theorem 7.3 . . . 173
F.4 Experimental details . . . 174

Bibliography **179**

Chapter

1 Introduction

De tous temps, les hommes ont voulu craquer le deep learning.

Adage populaire

Chapter content

1.1 Curse of dimensionality and structure in computer vision 2
1.1.1 Learning models of high-dimensional probability distributions 2
1.1.2 The curse of dimensionality in supervised learning 4
1.1.3 The curse of dimensionality in unsupervised learning 5
1.1.4 Deep convolutional neural networks 5
1.1.5 Leveraging structure to escape the curse of dimensionality 7
1.2 Properties of wavelet conditional probability distributions 10
1.2.1 Score-based diffusions and autoregressive factorizations 10
1.2.2 Conditional log-concavity of physical fields 13
1.2.3 Conditional locality and regularity of natural images 15
1.3 Non-linear operators for image classification 16
1.3.1 Separation and concentration in deep networks 17
1.3.2 Concentration with thresholdings in sparse representations 18
1.3.3 Separation with phase collapses of wavelet coefficients 20
1.4 A model of network weights with aligned random features 22
1.4.1 Random-feature kernels in deep networks 22
1.4.2 Evolution of kernels and training dynamics 23
1.4.3 Alignment convergence: the rainbow model 24
1.5 Organization of the dissertation 27

Deep neural networks have achieved considerable success in machine learning applications in the past ten years (LeCun et al., 2015). They have been applied to various types of data, including images, videos, audio data, time series, but also text, graphs, and game states, in both supervised and unsupervised learning tasks.

However, the fundamental principles behind these impressive results remain not well understood. In particular, the high-dimensionality of the data is a major challenge in theoretical analyses. It manifests itself in several forms, and leads to issues in function approximation, parameter estimation, generalization, and data generation. These challenges are collectively referred to as the *curse of dimensionality*.

In practice, deep learning seemingly escapes this curse and has emerged as an empirical solution to these challenges. Its success thus evidences a rich mathematical structure underlying real-world learning problems. This structure is revealed by the interplay between the training data (what properties are we implicitly relying on?), the architecture (what is the functional role

of network computations?), and the optimization algorithm (what has the network learned?). This dissertation studies these questions, in the context of convolutional networks applied to image generative modeling and image classification.

Outline. In this introduction, we present the various concepts used in the dissertation, as well as our contributions in relation to the prior state of the art.

In Section 1.1, we contrast the theoretical challenges coming from the curse of dimensionality in machine learning with the empirical success of deep-learning approaches. We then present classical notions of structure in the learning problem that can be leveraged to escape the curse. The next three sections tackle three different aspects of this structure: in the training data, in network computations, and in network weights.

In Section 1.2, we focus on the unsupervised learning problem of image generative modeling. We explain that a multiscale factorization of image distributions can reveal properties of conditional log-concavity, regularity, and local Markov dependency structure, whereas the global distribution does not enjoy these properties. These properties can then be leveraged to alleviate the curse of dimensionality.

In Section 1.3, we then turn to the supervised learning problem of image classification. We investigate the role of non-linearity in image classification, and review two classical interpretations which respectively leverage sparsity and symmetry groups. We show that the main function of the non-linearity in deep network classifiers is to collapse the phase of complex wavelet coefficients of network activations. In contrast, previous models based on thresholding and sparsity assumptions are neither sufficient nor necessary to explain the classification accuracy of deep networks.

In Section 1.4, we study properties of the weights of trained deep networks. We introduce a hierarchical kernel description of the network based on multi-layer random features. It leads to a probabilistic model of the learned weights. The model can be estimated from the weights of one or several trained networks, and allows generating new weights which can reach comparable accuracies without training.

Finally, we describe the organization of the dissertation in Section 1.5.

1.1 Curse of dimensionality and structure in computer vision

General machine learning problems suffer from the curse of dimensionality. In contrast, deep learning seems to escape this curse in typical computer vision tasks. This raises the following question: what kind of structure in image distributions are deep convolutional neural networks leveraging? In this section, we introduce the learning tasks studied in this dissertation (Section 1.1.1), their theoretical challenges (Sections 1.1.2 and 1.1.3), deep-learning approaches and their empirical results (Section 1.1.4), and classical assumptions on the structure of the data distribution to alleviate the curse of dimensionality (Section 1.1.5).

1.1.1 Learning models of high-dimensional probability distributions

Data distribution. Consider a probability distribution p known implicitly through a dataset of i.i.d. samples. In this dissertation, our data will consist in sets of images coming from several sources, ranging from physical fields to natural images, illustrated in Figure 1.1. In addition to distributions $p(x)$ of images x, we shall also consider joint distributions $p(x, y)$ of both images x and class labels y (e.g., "car" or "ship").

Approximation in a parametric model. A major goal in science is then to learn a parametric model of p from the training dataset. In unsupervised learning, we wish to model the whole

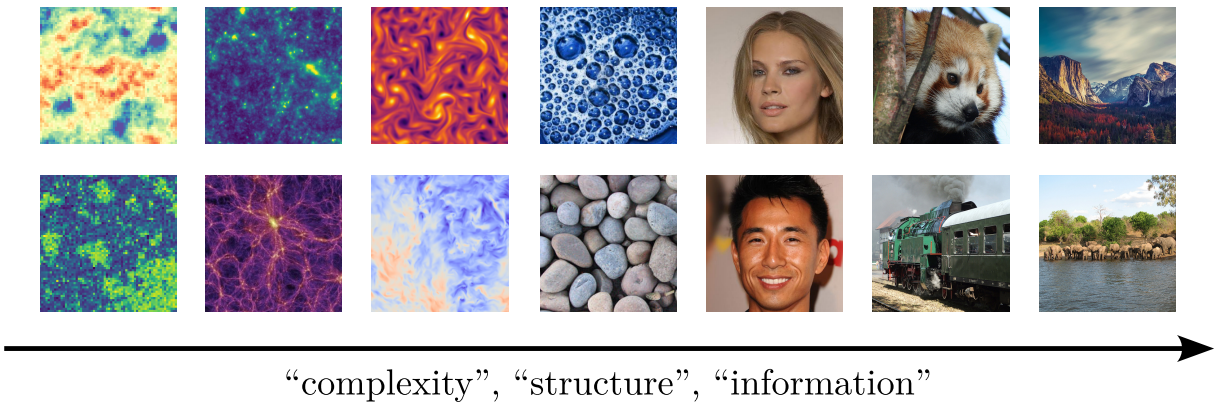

FIGURE 1.1: The image distributions typically considered in machine learning can be informally organized on a "complexity" axis, which measures the amount of "structure" or "information" in the image content (we use these words in an informal sense and do not imply any connections to related mathematical concepts). The axis ranges from toy theoretical models such as Gaussian processes, to physical fields such as turbulent flows, to natural textures, to photographic images such as human faces or natural scenes.

distribution $p(x)$, while in supervised learning we only care about the conditional distribution $p(y|x)$. This is done by introducing a parametric family such as an energy-based model

$$p_\theta(x) = \frac{1}{Z_\theta} \mathrm{e}^{-E_\theta(x)}, \qquad p_\theta(y|x) = \frac{1}{Z_\theta(x)} \mathrm{e}^{-E_\theta(y|x)}, \tag{1.1}$$

where E_θ is the "energy" and Z_θ is a normalizing factor so that $\int p_\theta(x)\mathrm{d}x = 1$ or $\int p_\theta(y|x)\mathrm{d}y = 1$ (for discrete labels y, the Lebesgue measure should be replaced with the counting measure).

Estimation of the parameters. Once the parametric model has been defined, the parameters θ need to be estimated from data so that the model p_θ is as close as possible to the data distribution p. This is usually quantified with the Kullback-Leibler divergence (or relative entropy) between these distributions, leading to the learning objective

$$\min_\theta \ \mathrm{KL}(p(x) \,\|\, p_\theta(x)), \qquad \min_\theta \ \mathbb{E}_{x\sim p}[\mathrm{KL}(p(y|x) \,\|\, p_\theta(y|x))]. \tag{1.2}$$

As p is only known through samples $x_1, \dots, x_n$ (for unsupervised learning) or $(x_1, y_1), \dots, (x_n, y_n)$ (for supervised learning), it is replaced by the empirical distribution of the training data, recovering the maximum likelihood principle

$$\min_\theta \frac{1}{n}\sum_{i=1}^{n} -\log p_\theta(x_i), \qquad \min_\theta \frac{1}{n}\sum_{i=1}^{n} -\log p_\theta(y_i|x_i). \tag{1.3}$$

Generation of samples from the model. Once the probability model has been learned, it can be used to evaluate the likelihood of a new data point x, or the conditional likelihood of candidate labels y given x. Another central task is then to draw samples from the modeled probability distribution, which generates new data points x (generative modeling) or predicted labels y given x (classification).

Unsupervised and supervised learning. Together, the unsupervised and supervised learning tasks can the be combined to obtain a model of and sample from the joint distribution $p(x, y)$. They thus represent complementary problems, but there is significant interaction between them. On the one hand, unsupervised learning problems are often turned into one or several supervised learning problems, as in self-supervised learning. On the other hand, supervised learning

problems can benefit from the analysis of their unsupervised counterparts. In particular, the model $p_\theta(y|x)$ can be expected to be accurate only when x is a typical sample of $p(x)$, so that the properties of $p(x)$ also play an important role in supervised learning.

However, a major difference between the two problems is the dimensionality of the considered objects. In supervised learning, the label y is typically low-dimensional or takes a small number of discrete values. In image classification, the number of classes is thus rarely above 10^3, and it is then feasible to enumerate all of them. In contrast, the image x is typically high-dimensional, with a dimensionality of the order of 10^6 for 512×512 color images. A coarse discretization of the space of possible images x already has a size of the order of 10^{10^6}, and brute force enumeration is then intractable. The probability distributions $p(x)$ or $p(y|x)$ are thus associated with different challenges, which we now present, starting with supervised learning.

1.1.2 The curse of dimensionality in supervised learning

The fact that y is low-dimensional or takes a small number of discrete values simplifies the tasks associated with the probability distribution $p(y|x)$. First, sampling from the conditional distribution $p_\theta(y|x)$ can be easily achieved by computing the conditional histogram of y. Second, computing the normalizing factor $Z_\theta(x) = \int \mathrm{e}^{-E_\theta(y|x)} \mathrm{d}y$ is also feasible. The remaining challenges associated with learning the model $p_\theta(y|x)$ are thus "only" the ones that arise in supervised learning of a function of the high-dimensional input x. We briefly enumerate these challenges, which arise from the classical bias-variance trade-off in balancing approximation and generalization error, and the computational hardness of general statistical estimation.

Approximation. Even though the label y is low-dimensional or discrete, it depends on the high-dimensional input x. The energy $E_\theta(y|x)$ is thus a function of the high-dimensional input (x, y). The approximation challenge consists in finding a parametric form of the energy $E_\theta(y|x)$ in eq. (1.1) that is expressive enough to capture the apparent complexity of the data presented in Figure 1.1. Without any prior information on the functional form of the true energy $E = -\log p$, the approximation class requires a number of parameters that is exponential in the dimensionality of x.

Estimation. The parameters θ are estimated by solving the optimization problem in eq. (1.3), or a variant of it. It requires to minimize a function of the high-dimensional parameter vector θ. This function is in general non-convex, so that (stochastic) gradient descent may be slowed down by saddle points or remain trapped in a local minimum. In general, finding the global minimum of the loss function then requires a time that is exponential in the dimensionality of θ.

Generalization. While optimization is performed on the empirical negative log-likelihood in eq. (1.3), we wish to control the Kullback-Leibler divergence with respect to the unknown distribution p in eq. (1.2). Without any assumptions on the data distribution, generalizing to unseen test data requires a number of training samples that is exponential in the dimensionality of x.

In this work. In this dissertation, we will mostly focus on the approximation challenge in the setting of image classification, though the estimation and generalization challenges are of course relevant in our numerical experiments.

1.1.3 The curse of dimensionality in unsupervised learning

In unsupervised learning, one wishes to learn and sample from a model of the probability distribution $p(x)$. As it is a function of the high-dimensional input x, the challenges presented above in Section 1.1.2 also apply here. However, the probability distribution $p(x)$ is now over the high-dimensional input x, rather than the low-dimensional or discrete label y as in $p(y|x)$. This leads to additional challenges which we now review.

Estimation. The log-likelihood of a data point x depends on the normalizing factor Z_θ. Z_θ is typically unknown and its computation is in general cursed by dimensionality as it requires computing an integral over the high-dimensional input x. This can also be seen by computing the gradient of the log-likelihood (the Fisher score)

$$-\nabla_\theta \log p_\theta(x) = \nabla_\theta E_\theta(x) - \mathbb{E}_{x\sim p_\theta}[\nabla_\theta E_\theta(x)], \tag{1.4}$$

which requires generating samples from the *model* p_θ. This generation task is also cursed by dimensionality, as we now explain.

Generation. In general, drawing a sample from a probability distribution given by its energy function $E_\theta(x)$ has a time complexity that is exponential in the dimensionality of x. By drawing a parallel with optimization, the problem of generation requires finding the regions of the space where the energy E_θ is close enough to its minimum. Without any prior information on the location of the modes of p_θ, one has no option but to explore the whole space, whose volume is exponential in the dimensionality of x. For instance, generic Markov Chain Monte Carlo algorithms need to wait for an exponential amount of time before a random proposal stumbles upon a given mode of p_θ.

In this work. In this dissertation, we will tackle both of these issues in the setting of image generative modeling.

1.1.4 Deep convolutional neural networks

Despite these negative theoretical results, deep learning has achieved surprising success in practice (LeCun et al., 2015). We briefly present its core components and its achievements in image classification and generative modeling.

Deep neural networks. An energy-based model must specify the functional form of $E_\theta(x)$. Neural networks specify this functional form through their architecture. A major characteristic is that it is compositional and cascades linear layers $(W_j)_{1\leq j\leq J}$ with pointwise non-linearities ρ such as rectified linear units (Nair and Hinton, 2010). The network function thus writes

$$E_\theta(x) = W_J \rho(W_{J-1}\rho(\cdots \rho(W_1 x)\cdots)), \tag{1.5}$$

with parameters $\theta = (W_1, \ldots, W_J)$. The network is usually optimized end-to-end with stochastic gradient descent using back-propagation (LeCun et al., 1989a). There are many hyper-parameters and modifications of this basic scheme that play an important role in practice, such as mini-batching, changing the optimization algorithm (Kingma and Ba, 2014), momentum, learning rate scheduling (Smith, 2017), data augmentation, batch-normalization (Ioffe and Szegedy, 2015), initialization scheme, or regularization such as weight decay.

There are other types of layers beyond the two above, most notably normalization layers such as batch-normalization or divisive normalization. Although they play an important role in numerical applications, we shall not focus on these layers and do not write them explicitly.

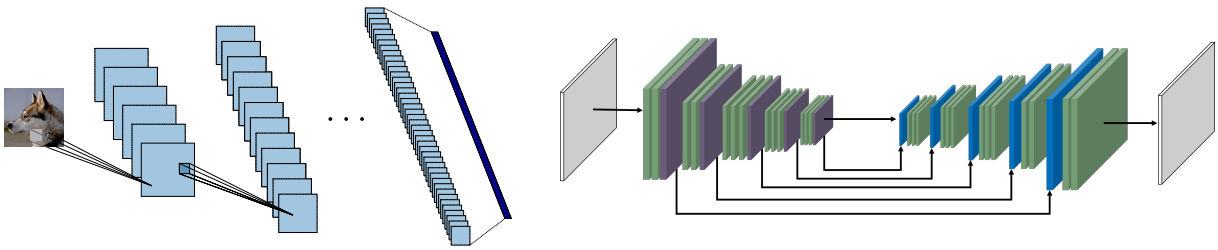

FIGURE 1.2: Examples of convolutional architectures. Left: *a convolutional classifier architecture, which progressively reduces the spatial resolution while increasing the number of channels.* Right: *a U-Net architecture as typically used in score-based diffusion models, which is a symmetric encoder-decoder architecture.*

Similarly, skip-connections have become an ubiquitous component in deep architectures (He et al., 2016). They can be incorporated in eq. (1.5) by augmenting the non-linearity ρ with the identity, so that the network non-linearity is $t \mapsto (\rho(t), t)$. We also mention attention layers, which are used extensively in transformer architectures (Vaswani et al., 2017; Dosovitskiy et al., 2021) that we shall not consider in this dissertation.

Convolutional architectures. In practice, the architecture of the network is further constrained by imposing additional structure on the linear layers. Convolutional architectures (LeCun et al., 1989a; LeCun and Bengio, 1995) impose that the linear layers W_j are convolutions with small filters. The weight sharing between different neurons at a given layer implies that the linear layers are equivariant to translations, while restricting the neuron receptive field size leads to locality properties.

Convolutional networks further introduce subsampling or pooling layers, which iteratively reduce the spatial resolution. Spatial dimensions are progressively transformed into channel dimensions. It leads to a hierarchical, multiscale architecture which is illustrated in the left panel of Figure 1.2. Such architectures are appropriate when the output is a scalar or a low-dimensional vector without spatial topology, as in $E_\theta(x)$ or $E_\theta(y|x)$.

In some cases, the desired output is an image, such as when directly modeling the Stein score $\nabla_x E_\theta(x)$ in score-based diffusion models. U-Nets (Ronneberger et al., 2015), also known as hourglass networks (Newell et al., 2016), include upsampling or transposed convolution layers that progressively increase the spatial resolution back to that of the input image x. The architecture is then composed of an encoder stage and a decoder stage, with skip-connections from the encoder to the decoder at each scale. The decoder combines global information computed by the encoder with fine spatial localization to reconstruct a high-resolution output. The architecture is illustrated in the right panel of Figure 1.2. Its computation graph resembles that of a forward and backward propagation in a network computing a scalar function, except that the weights of the encoder and the decoder are not tied.

Empirical success. Deep-learning-based approaches have obtained impressive results on a variety of tasks.

In image classification, the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015) has spurred a dramatic improvement in accuracies over the years as the number of network layers increased (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016), as shown in the left panel of Figure 1.3. Nowadays, the state-of-the-art performance is obtained with transformer architectures (Dosovitskiy et al., 2021) reaching 91% top-1 accuracies with billions of parameters, though purely convolutional architectures have been shown to achieve comparable accuracies (Liu et al., 2022b). Though improvements are in general attributed to the architecture, they also result from increases in data quantity or quality (Hinton et al., 2015), computing power (Sutton, 2019), and optimization recipes (Wightman et al., 2021).

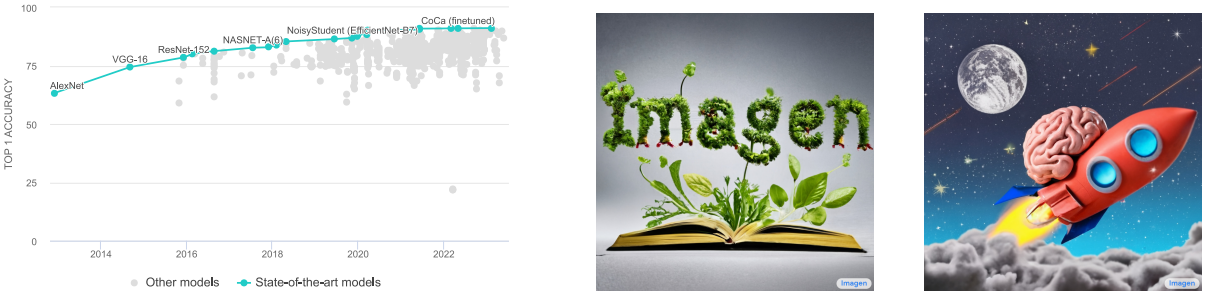

FIGURE 1.3: Achievements of deep learning in computer vision. Left: *state-of-the-art top-1 accuracies on the ImageNet dataset over the years (source: https://paperswithcode.com/sota/image-classification-on-imagenet).* Right: *images generated by the Imagen score-based diffusion model (Saharia et al., 2022) conditioned by a text caption (reproduced from the original publication).*

In image generative modeling, diffusion models (Sohl-Dickstein et al., 2015) and score-based generative models (Song and Ermon, 2019; Song et al., 2021b) have obtained state-of-the-art results (Dhariwal and Nichol, 2021). These two approaches are essentially equivalent (Ho et al., 2020), and we shall refer to them generally as score-based diffusion models in this dissertation. Score-based diffusion models estimate the (Stein) scores of the probability distributions of the image x and its contaminations with Gaussian white noise of any variance. This is typically achieved by a convolutional U-Net architecture augmented with attention layers at coarse resolutions. Score-based diffusion models can be combined with language models to generate images conditioned on a caption (Saharia et al., 2022; Ramesh et al., 2022; Rombach et al., 2022). The quality of the high-resolution generated images is impressive, as shown in the right panel of Figure 1.3.

In this work. These results come in contrast with the theoretical challenges outlined in Sections 1.1.2 and 1.1.3. They imply that network architectures and training algorithms are adapted to image distributions, and implicitly rely on some hidden properties. What are these properties, and how can they be leveraged by deep networks to achieve low error? We present in the next section several such properties that have been studied in the literature.

1.1.5 Leveraging structure to escape the curse of dimensionality

We briefly review some classical structural assumptions that allow alleviating the curse of dimensionality. The properties considered below correspond to assumptions on the energy function E associated to the data probability distribution p, with $E(x) = -\log p(x)$ or $E(y|x) = -\log p(y|x)$. They thus apply to both the training data distribution and the functional form of the target function to approximate. This list is not meant to be exhaustive, nor a review of theoretical results. Rather, it is an exposition of concepts used in the remainder of this dissertation.

Locality and low-dimensionality. One can assume that the energy function E is local, in the sense that it decomposes as a sum of functions that only depend on pixel values inside small patches of the image x. It defines an approximation class where learning is reduced to lower-dimensional functions, thus improving approximation and generalization performance. In generative modeling, this locality assumption is equivalent to assuming a Markov random field model (Geman and Geman, 1984) due to the Hammersley-Clifford theorem (Clifford and Hammersley, 1971). In classification, this leads to a conditional Markov random field (Lafferty et al., 2001) which recovers the naive Bayes classifier in the extreme case of non-overlapping patches comprised of individual pixels.

One can generalize this assumption by replacing subsets of image pixels with general low-dimensional projections of x, which then play the role of latent variables. These latent variables can be either known or unknown, in which case they have to be learned (but assuming their existence may still alleviate the curse of dimensionality). The spatial topology of images however provides a strong inductive bias towards latent variables that are localized in the spatial and/or frequency domains.

Combinations of such hypotheses have been used with various names in the literature, such as generalized additive models, non-parametric ANOVA models, projection pursuit, or multi-index models (Bach, 2017a).

Multiscale structure and compositionality. Rather than added together, local functions in the sense of the above paragraph can also be composed together. This compositionality assumption implies that the energy function E admits a hierarchical form where local information is iteratively processed and then aggregated over more global regions. The constituent functions are then all low-dimensional, alleviating the curse of dimensionality for the approximation (Poggio et al., 2017) and generalization errors (Li et al., 2021) in some settings.

Such assumptions of hierarchical locality are central in computer vision (Burt and Adelson, 1983; Mallat, 2008) and closely mirror the architecture of convolutional neural networks. The axis of depth inside the network then corresponds to an axis of spatial scale. This compositionality property then assumes that the energy E has a local multiscale structure, in the sense that it can be computed with local processing at each scale.

Stationarity and symmetry groups. Another assumption that is commonly used in computer vision alongside multiscale structure is stationarity. It states that the energy function E is invariant with respect to translations of the image x, i.e., an image or its translation are equally likely to be observed (for $E(x)$), and they have the same label y (for $E(y|x)$). This assumption motivates the use of weight sharing in convolutional neural networks and thus reduces their parameter count.

This concept can be generalized to other groups, including geometric groups such as rotations or scalings. In particular, scaling-invariance implies self-similarity properties on the data probability distribution. Functions that are invariant to these group actions are similarly obtained in a hierarchical manner by cascading group-equivariant functions such as group convolutions (Anselmi et al., 2015; Cohen and Welling, 2016; Kondor and Trivedi, 2018). Such approaches reduce the dimensionality of the approximation class by the dimensionality of the group (Anselmi et al., 2016; Mei et al., 2021; Bietti et al., 2021), so that larger groups are needed for larger improvements.

One such large group is the group of diffeomorphisms, acting on images by deformation. Its infinite dimensionality prevents the computation of exact invariants, and the assumption is then that the energy function E is regular with respect to its action. The scattering transform (Mallat, 2012) is Lipschitz with respect to the action of diffeomorphisms, and is thus approximately invariant to deformations.

Smoothness and kernels. One may consider different regularity properties than those arising from group actions. An important instance of such assumptions is that the energy function E belongs in a specific reproducing kernel Hilbert space (RKHS) (Schölkopf and Smola, 2002). It corresponds to a smoothness prior, as the RKHS norm of E controls its Lipschitz constant with respect to the geometry defined on x by the associated kernel. The energy function E can then be modeled as a linear function over a fixed, possibly infinite-dimensional feature map corresponding to the chosen kernel. The approximation class takes the form of an exponential family, which is linear in the parameters and leads to a convex optimization problem (recovering logistic regression in the supervised setting).

The estimation of the optimal parameters can be solved exactly by leveraging the dual formulation, or approximately with random-feature expansions of the infinite-dimensional kernel feature map. The approximation and generalization properties of random-feature approximations are now rather well known (Rahimi and Recht, 2008; Bach, 2017b; Rudi and Rosasco, 2017; Mei et al., 2022). One then obtains polynomial error rates that are independent of the dimensionality of the input, albeit with an added "implicit" ridge regularization (Jacot et al., 2020). The key quantities which determine these error rates are the spectrum of the covariance matrix of the feature map, and the coefficients of the energy function in its eigenbasis (Caponnetto and De Vito, 2007). A fast decay of both the covariance spectrum and the target coefficients leads to faster rates, which shows that random feature approximations enjoy a (limited) adaptivity to the problem characteristics.

Sparsity of weights and representations. While kernel methods perform implicitly or explicitly an ℓ^2 regularization, ℓ^1 regularization has also been classically studied, under the names of lasso (Tibshirani, 1996) and basis pursuit (Chen et al., 2001). Though estimation is more computationally challenging, ℓ^1 regularization enjoys better approximation and generalization properties by being adaptive to low-dimensional latent variables (Bach, 2017a), contrarily to generic kernel methods. The ℓ^1 penalty on the parameters corresponds to a variation norm of the associated function, and the resulting approximation space can be characterized as a reproducing kernel Banach space (Bartolucci et al., 2021). The minimization of this variation norm also arises as the implicit bias of one-hidden-layer networks in some supervised classification tasks (Chizat and Bach, 2020).

Rather than assuming a sparse prior on the parameters, one can also assume that the input data admits a sparse representation, where the ordered amplitudes of the representation coefficients have a fast decay. Eliminating the smallest coefficients then defines efficient adaptive low-dimensional approximations, thus alleviating the curse of dimensionality. Sparse representations have a long history in computer vision (Mallat, 2008), from fixed curvelet tight frames (Candès et al., 1999) or adaptive bandlet bases (Le Pennec and Mallat, 2005) to sparse coding in learned redundant dictionaries (Olshausen and Field, 1997; Elad and Aharon, 2006).

Log-concave distributions. The assumptions outlined above are mostly targeting the approximation, generalization, and sometimes estimation errors in supervised learning. As explained in Section 1.1.3, unsupervised learning faces additional challenges when estimating parameters of or generating samples from high-dimensional probability distributions. The curse of dimensionality can be lifted if the data probability distribution p is log-concave, or equivalently, if the energy function E is convex.

First, maximum-likelihood estimation in a log-concave parametric class can be relaxed (Koehler et al., 2022) to score matching (Hyvärinen and Dayan, 2005), which avoids the need to compute normalizing factors. Second, log-concavity also allows escaping the curse when generating samples. For instance, the Metropolis-adjusted Langevin algorithm enjoys in this case convergence guarantees that are polynomial in the dimensionality of x (Chewi, 2023) with an algorithmic warm-start (Altschuler and Chewi, 2023).

In this work. We have presented several classical assumptions that have been used in the literature to alleviate the curse of dimensionality. This leads to two questions: can they be used to explain and understand the performance of deep convolutional neural networks? To what extent does explicitly relying on these sources of structure allow recovering their performance? In this dissertation, we will use numerical experiments to both investigate properties of deep networks and evaluate the accuracy of constrained deep architectures.

The next sections present our results in various contexts, through the lens of the training data (what are its properties?), the architecture (what is the role of its computations?), and

the optimization algorithm (what have the weights learned?). First, we show in Section 1.2 that image distributions can exhibit properties of log-concavity, smoothness, and locality when they are factorized with a multiscale decomposition. Second, we present in Section 1.3 investigations on the role of the non-linearity in image classification architectures, showing that it mainly computes invariants to spatial deformations with phase collapses. Third, we introduce in Section 1.4 a probabilistic model of the learned weights of deep networks based on hierarchical random-feature kernels.

1.2 Properties of wavelet conditional probability distributions

In this section, we show that image probability distributions can be factorized as a produce of wavelet conditional distributions that are log-concave, smooth, and local, whereas the global image distribution does not enjoy these properties. These properties can be leveraged to alleviate the curse of dimensionality in image generative modeling.

We begin by explaining in Section 1.2.1 how both score-based diffusions and autoregressive factorizations manage to reduce generative modeling to supervised learning. This is achieved by decomposing the probability distribution of x into conditional distributions with tractable estimation and generation but possibly intractable approximation and generalization. This important observation implies that these four aspects should be considered together.

We introduce in Section 1.2.2 a framework in which all the above challenges can be tackled simultaneously. It builds upon the prior work of Marchand et al. (2022), who showed that a class of multiscale physical fields can be modeled with local conditional distributions at each scale, thus taking care of the approximation and generalization errors. We show that these conditional distributions are log-concave, which allows controlling the estimation and generation errors.

In Section 1.2.3, we show that the wavelet conditional distributions considered by Marchand et al. (2022) also enjoy some of these properties in the case of natural images. We demonstrate empirically that this factorization can be incorporated in score-based diffusions to achieve a linear sampling complexity. It provides a theoretical justification for this approach, used extensively in the literature. We further show that the distribution of face images, which is non-stationary and has large-scale structure, can be approximated with stationary and local *conditional* distributions at each scale. It leads to lower-dimensional score models that can be approximated with networks with smaller receptive fields.

1.2.1 Score-based diffusions and autoregressive factorizations

We have presented the challenges posed by the curse of dimensionality in Sections 1.1.2 and 1.1.3. The "unsupervised"-type issues arise when estimating normalizing factors and generating samples from probability distributions over high-dimensional variables. They can be tackled by decomposing the image distribution $p(x)$ into probability distributions that are either *log-concave* or *over low-dimensional variables*. The "supervised"-type issues arise when learning approximations of high-dimensional functions. They can be tackled when prior information allows specifying *low-dimensional parametric models* of these distributions.

We explain in this section that the first type of issues can be solved without any prior assumptions. This is achieved by two approaches: score-based diffusions, which leverage log-concave decompositions, and autoregressive factorizations, which leverage low-dimensional decompositions. In effect, they provide a reduction of unsupervised learning to supervised learning. As a result, the challenges of Sections 1.1.2 and 1.1.3 should be considered together when studying the curse of dimensionality in generative modeling.

We begin by reminding that log-concavity allows escaping the curse through a log-Sobolev inequality. We then explain that score-based diffusion models leverage a similar property through

a higher-dimensional lifting to distributions over paths. Finally, we highlight that an autoregressive factorization also succeeds in escaping this curse, by implicitly assuming that the conditional distributions can be accurately approximated.

Log-Sobolev inequality and log-concavity. The error introduced by modeling the true data distribution $p(x) = \mathrm{e}^{-E(x)}$ with the energy-based model $p_\theta(x) = Z_\theta^{-1}\mathrm{e}^{-E_\theta(x)}$ can be quantified with the Kullback-Leibler divergence

$$\mathrm{KL}(p \,\|\, p_\theta) = \mathbb{E}_p[E_\theta(x) + \log Z_\theta - E(x)] = \mathbb{E}_p[E_\theta(x)] + \log Z_\theta + \mathrm{cst}, \tag{1.6}$$

or the Fisher divergence[1]

$$\mathrm{FI}(p \,\|\, p_\theta) = \mathbb{E}_p\left[\frac{1}{2}\|\nabla E_\theta(x) - \nabla E(x)\|^2\right] = \mathbb{E}_p\left[\frac{1}{2}\|\nabla E_\theta(x)\|^2 - \Delta E_\theta(x)\right] + \mathrm{cst}. \tag{1.7}$$

The latter does not depend on the normalizing factor Z_θ and can thus be computed efficiently (up to a constant). For exponential families, eq. (1.7) is even a quadratic function of the parameters and can be minimized in closed form.

The Fisher divergence is however weaker than the Kullback-Leibler divergence, as quantified by the log-Sobolev inequality (Gross, 1975; Markowich and Villani, 2000)

$$\mathrm{KL}(p \,\|\, p_\theta) \leq \frac{1}{\rho_\theta}\,\mathrm{FI}(p \,\|\, p_\theta), \tag{1.8}$$

where ρ_θ is the log-Sobolev constant of p_θ. In general, ρ_θ decreases exponentially with the dimensionality of x, but it is controlled when p_θ is strongly log-concave, i.e., E_θ is strongly convex (Bakry et al., 2014). A non-vanishing log-Sobolev constant implies that both parameter estimation and sample generation can be solved efficiently.

First, eq. (1.8) shows that score matching leads to guarantees in the Kullback-Leibler divergence. Its statistical efficiency is thus comparable to maximum-likelihood estimation (Koehler et al., 2022) while being much lighter computationally.

Second, eq. (1.8) also implies an exponential convergence of the Langevin diffusion

$$\mathrm{d}x_t = -\nabla E_\theta(x_t)\mathrm{d}t + \mathrm{d}w_t \tag{1.9}$$

where (w_t) is a Wiener process. Indeed, a direct calculation shows that

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{KL}(p_t \,\|\, p_\theta) = -\,\mathrm{FI}(p_t \,\|\, p_\theta) \leq -\rho_\theta\,\mathrm{KL}(p_t \,\|\, p_\theta), \tag{1.10}$$

where p_t is the distribution of x_t, which then converges exponentially fast towards p_θ at a rate at least ρ_θ (Markowich and Villani, 2000). With an appropriate discretization of eq. (1.9), it leads to a polynomial sampling complexity (Chewi, 2023; Altschuler and Chewi, 2023).

Score-based diffusion models. Rather than modeling the probability distribution of x, score-based diffusions build a model of sample paths $(x_t)_{t\in[0,T]}$ of a forward "noising" process: for instance, starting from $x_0 = x$, define

$$\mathrm{d}x_t = \mathrm{d}w_t, \tag{1.11}$$

where $(w_t)_t$ is a Wiener process, so that we have the conditional distributions

$$x_t \,|\, (x_0, \ldots, x_{t-\delta}) \sim \mathcal{N}(x_{t-\delta}, \delta\,\mathrm{Id}). \tag{1.12}$$

[1] Our notation differs from the conventional use by a factor 1/2 for convenience.

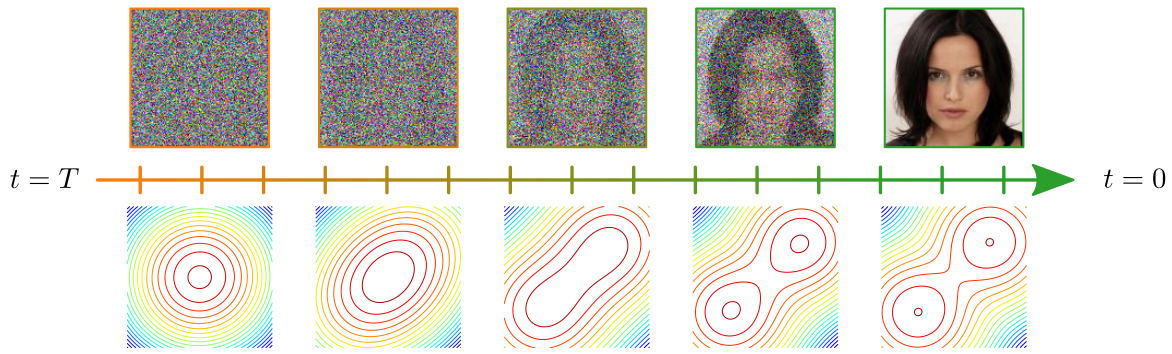

FIGURE 1.4: The backward "denoising" process of a score-based diffusion model maps white Gaussian noise to the data probability distribution. The non-log-concave distribution is represented by its scores along the diffusion path, by encoding relative amplitudes between disconnected modes at times where those modes merge together.

Crucially, the law of the process $(x_t)_t$ is also described by a backward "denoising" process,

$$\mathrm{d}x_t = -\nabla E_t(x_t)(-\mathrm{d}t) - \mathrm{d}w'_t, \tag{1.13}$$

where $(w'_t)_t$ is a backward Wiener process (Anderson, 1982) and $E_t = -\log p_t$ is the energy associated to the marginal distribution of x_t in eq. (1.11). The backward process, illustrated in Figure 1.4, is started from $x_T \sim p_T \approx \mathcal{N}(0, T\,\mathrm{Id})$ (for large T) and depends on the scores $(\nabla E_t)_t$. It means that when the stepsize δ vanishes,

$$x_{t-\delta} \,|\, (x_t, \dots, x_T) \underset{\delta \to 0}{\sim} \mathcal{N}(x_t - \delta \nabla E_t(x_t), \delta\,\mathrm{Id}). \tag{1.14}$$

The joint distribution of (x_t) is thus factorized as an infinite Markov product (over continuous time $t \in [0, T]$) of white Gaussian distributions.

A score-based diffusion model is defined by energy-based models $(E_{\theta,t})_t$ at each time t. Let p_θ be the distribution of x_0 obtained from the backward diffusion

$$\mathrm{d}x_t = -\nabla E_{\theta,t}(x_t)(-\mathrm{d}t) - \mathrm{d}w'_t, \tag{1.15}$$

started from $x_T \sim \mathcal{N}(0, T\,\mathrm{Id})$. Finally, let $\tilde{p}$ and $\tilde{p}_\theta$ respectively be the joint distributions of the sample paths $(x_t)_t$ of eqs. (1.13) and (1.15). Then the data-processing inequality and a direct calculation (Song et al., 2021a) imply that

$$\mathrm{KL}(p \,\|\, p_\theta) \le \mathrm{KL}(\tilde{p} \,\|\, \tilde{p}_\theta) = \int_0^T \mathrm{FI}(p_t \,\|\, p_{\theta,t})\mathrm{d}t + o(\mathrm{e}^{-T}), \tag{1.16}$$

which plays a similar role as the log-Sobolev inequality, but where the Fisher divergence has to be considered *at all times* (compare eq. (1.16) with eq. (1.8)). As suggested by eq. (1.14), score-based diffusion models thus automatically enjoy properties similar to log-concavity after the lifting to distributions over paths.

First, eq. (1.16) provides a control on the model error in Kullback-Leibler divergence from the score-matching error at all times, hence bypassing the issues associated with normalizing factors.

Second, eq. (1.15) inherits the exponential convergence of eq. (1.11), and Chen et al. (2022b,a) have shown that it can be discretized without blowing up score errors, leading to a total error which is polynomial in the number of iterations, score matching loss, and data dimensionality. Diffusion models thus reduce the problem of generative modeling of a distribution p to that of estimating the scores $(\nabla E_t)_t$, which amounts to denoising (i.e., self-supervised learning of a high-dimensional function) through the common denoising score matching formulation (Vincent,

2011; Kadkhodaie and Simoncelli, 2021). It shows that generative modeling is at least as easy as learning the scores, or equivalently, that *learning the scores is at least as hard as generative modeling*.

Though deep score networks have been shown to generate images of extremely high quality, the extent to which they truly learn the scores of the data distribution is unclear, for two reasons. First, practitioners have empirically optimized generation quality and efficiency rather than data fidelity, and minimize a reweighted version of eq. (1.16) which puts less emphasis on smaller times, leading to over-smoothed generated images (Kingma and Gao, 2023; Karras et al., 2022). Smaller times are typically harder to model but capture the fine details of the data probability distribution, which have empirically been found to dramatically alter the expected model log-likelihood (Nichol and Dhariwal, 2021; Song et al., 2021a). Second, recent work has evidenced that these deep score networks can memorize their training data (Carlini et al., 2023; Somepalli et al., 2022), indicating a lack of generalization.

It is therefore of fundamental importance to understand the properties which allow learning accurate score models. In this dissertation, we show as a step towards this goal that a multiscale factorization of the probability distribution can lead to local and therefore low-dimensional score models.

Autoregressive factorization. Another way to deal with the curse of dimensionality is to reduce the dimensionality of the random variables. This can be achieved with a factorization of the probability distribution of $x \in \mathbb{R}^d$ as products of one-dimensional conditional distributions

$$p(x) = p(x[1]) \prod_{i=2}^{d} p(x[i] \,|\, x[1], \dots, x[i-1]). \tag{1.17}$$

This autoregressive factorization, which is classical in time series modeling, has also been used to model images in deep learning (Van Den Oord et al., 2016). One is thus faced with a sequence of one-dimensional *conditional* generative modeling problems, akin to supervised learning of the conditional energy $E(y|x)$.

First, a model $p_\theta(x)$ is obtained with models $p_{\theta_i}(x[i] \,|\, x[1], \dots, x[i-1])$, for which normalizing factors can be straightforwardly computed because $x[i]$ is one-dimensional.

Second, a sample x from $p_\theta(x)$ can be generated iteratively by first sampling the first component $x[1]$ from $p_{\theta_1}(x[1])$, and iteratively the i-th component $x[i]$ from $p_{\theta_i}(x[i] \,|\, x[1], \dots, x[i-1])$.

This approach has the advantage of breaking the curse of dimensionality, *provided that one can learn models of the conditional factors* $p(x[i] \,|\, x[1], \dots, x[i-1])$. These conditional factors are functions of high-dimensional inputs due to the a priori dependence between the components of x. Indeed, the conditional density may have a much more complicated functional form than the joint distribution $p(x)$. Equation (1.17) can still be used if it is known a priori that the conditional distributions $p(x[i] \,|\, x[1], \dots, x[i-1])$ assume a "simple" functional form. For instance, Markov random fields (Geman and Geman, 1984) assume conditional independence properties between components of x, leading to local and thus low-dimensional conditional densities, as explained in Section 1.1.5. Such models assume that long-range *dependencies* arise from short-range *interactions*. This assumption is however too restrictive to model complex image distributions that can have long-range interactions.

In this dissertation, we rather consider factorizations obtained by conditioning over *spatial scale* rather than spatial position, leading to local interactions at each scale.

1.2.2 Conditional log-concavity of physical fields

We have explained in Section 1.2.1 that it is necessary to take into account the approximation and generalization challenges together with the estimation and generation challenges that are specific to unsupervised learning. Lifting the curse of dimensionality thus requires a factorization

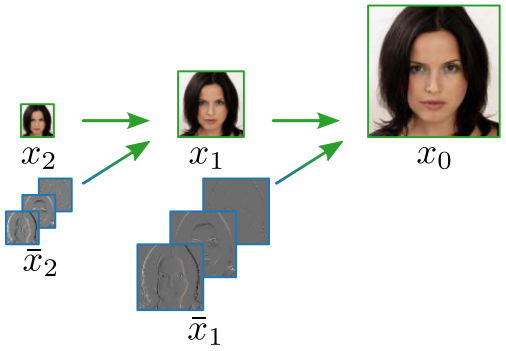

FIGURE 1.5: The wavelet conditional factorization of Marchand et al. (2022) generates a high-resolution image by generating a coarse approximation and then conditionally generating detail coefficients in the wavelet domain.

into probability distributions that are either log-concave or over low-dimensional variables, *and admit low-dimensional parametric models*. The wavelet conditional factorization of Marchand et al. (2022) has been shown to provide such low-dimensional parametric models of multiscale physical fields. We build upon it by additionally demonstrating that the wavelet conditional distributions are log-concave. This latter property allows controlling simultaneously all sources of errors when combined with the former one.

Wavelet conditional factorization. An image $x = x_0$ can be decomposed with a wavelet transform as "detail coefficients" $(\bar{x}_j)_{j \leq J}$ at each scales and a coarse approximation x_J of x at the largest scale 2^J. The wavelet coefficients $(\bar{x}_{j+1}, \bar{x}_{j+2}, \dots, \bar{x}_J)$ and the coarse approximation x_J can be combined to define an approximation x_j of x at the scale 2^j. The wavelet conditional factorization introduced by Marchand et al. (2022) writes

$$p(x) = p(\bar{x}_1, \dots, \bar{x}_J, x_J) = p(x_J) \prod_{j=1}^{J} p(\bar{x}_j \,|\, \bar{x}_{j+1}, \dots, \bar{x}_J, x_J) = p(x_J) \prod_{j=1}^{J} p(\bar{x}_j \,|\, x_j). \tag{1.18}$$

Such a factorization means that one can first generate a sample at a coarse resolution and then perform "generative upsampling" iteratively by conditionally generating details. This is illustrated in Figure 1.5.

This factorization has two desirable properties. First, if the image distribution $p(x)$ satisfies a self-similarity property, then the conditional distributions over scales $p(\bar{x}_j|x_j)$ have a similar functional form, as they are directly related to the marginal coarser-scale distributions $p(x_{j-1})$. This idea is at the heart of the renormalization group in statistical field theory (Wilson, 1971), which inspired the probability factorization of Marchand et al. (2022). Second, long-range dependencies may be more efficiently represented as a short-range interactions *at each scale*, leading to a cascade of conditional Markov random fields rather than a joint Markov random field, as done by Marchand et al. (2022). The probability distribution $p(\bar{x}_j|x_j)$ encode the interactions between wavelet coefficients at different scales, but represented at the same spatial resolution. Their short-range interactions can thus thus be likened to operators that apply along channels inside a convolutional neural network.

However, the factors $p(\bar{x}_j|x_j)$ are probability distributions over high-dimensional variables, and thus suffer from issues with the estimation of normalizing factors and generation of samples. We now explain that these factors can be made log-concave by generalizing the probability decomposition.

Conditionally log-concave factorizations. We introduce in Chapter 2 a generalization of both eqs. (1.17) and (1.18) with arbitrary orthogonal projectors. We prove that if each con-

ditional distribution $p(\bar{x}_j|x_j)$ is conditionally log-concave, then one obtains both learning and sampling algorithms with a polynomial complexity, provided that the conditional distributions can be modeled with low-dimensional exponential families. Combined with the multiscale locality shown in Marchand et al. (2022), it provides a complete control over all sources of error.

Conditional log-concavity arises when the energy $E(x)$ is dominated by quadratic interactions, in the following sense: write

$$E(x) = \frac{1}{2}x^{\mathrm{T}}Kx + V(x), \tag{1.19}$$

with K a positive symmetric matrix representing the "kinetic energy" and V is a possibly non-convex non-quadratic function representing the "potential energy". The log-concavity of p is equivalent to the convexity of E, or equivalently to the condition $\nabla^2 E(x) = K + \nabla^2 V(x) \succcurlyeq 0$. The large eigenvectors of K thus define directions where the energy is a priori "more convex". For multiscale stationary image distributions, K is a convolution whose eigenvalues have a power-law growth at high frequencies.

It is proved in Chapter 2 that the φ^4 energy from statistical field theory is indeed convex over a small-enough high-frequency band when conditioned on the remaining lower-frequency band. A log-concave conditional distribution $p(\bar{x}_1|x_1)$ can then be obtained by considering wavelet packet projectors, which define a narrower frequency decomposition $x = x_0 \mapsto (\bar{x}_1, x_1)$ than the dyadic splitting of the wavelet transform. The argument can then be iterated by replacing $p(x_0)$ with $p(x_1)$ and exploiting the self-similarity over scales of the φ^4 energy at the critical temperature. We further demonstrate numerically that cosmological weak-lensing images (Bartelmann and Schneider, 2001; Kilbinger, 2015) also have a conditionally log-concave distribution. It shows that complex non-log-concave distributions $p(x)$ may still be written as *a product of* log-concave conditional distributions.

Contributions. This approach provides an efficient generative modeling algorithm where all sources of errors are explicitly controlled. It lifts the curse of dimensionality for multiscale physical fields, where prior information guarantees conditional log-concavity and provides low-dimensional parametric models. This is a promising first step towards defining more general classes of probability distributions that could apply to natural images.

1.2.3 Conditional locality and regularity of natural images

In the more challenging setting of natural images, it is less clear that the conditional distributions $p(\bar{x}_j|x_j)$ are log-concave or local. However, score-based diffusion models still rely on a multiscale iterative approach similar to Marchand et al. (2022) to generate high-resolution images (Saharia et al., 2021; Ho et al., 2022; Dhariwal and Nichol, 2021). We explain that the multiscale factorization enables the use of local score networks with limited receptive fields, thus alleviating the curse of dimensionality. Additionally, it leads to a reduced sampling complexity from quadratic to empirically linear in the image dimensionality.

Wavelet score-based diffusion models. We introduce in Chapter 3 wavelet score-based diffusion models (referred to as wavelet score-based generative models in the main text). They are obtained by first factorizing the probability distribution $p(x)$ over scales as in eq. (1.18). Each conditional factor $p(\bar{x}_j|x_j)$ is then approximated with a *conditional* score-based diffusion model. A wavelet score-based diffusion model thus estimates the scores of noisy wavelet coefficients conditioned on clean low-resolution images. The probability distribution $p(x_J)$ is low-dimensional and is also approximated with a score-based diffusion model. The model is illustrated in Figure 1.6. It is equivalent to the iterative approaches of Saharia et al. (2021); Ho et al. (2022); Dhariwal and Nichol (2021), but it relies explicitly on the wavelet conditional distributions $p(\bar{x}_j|x_j)$ as opposed to the degenerate conditional distributions $p(x_{j-1}|x_j)$.

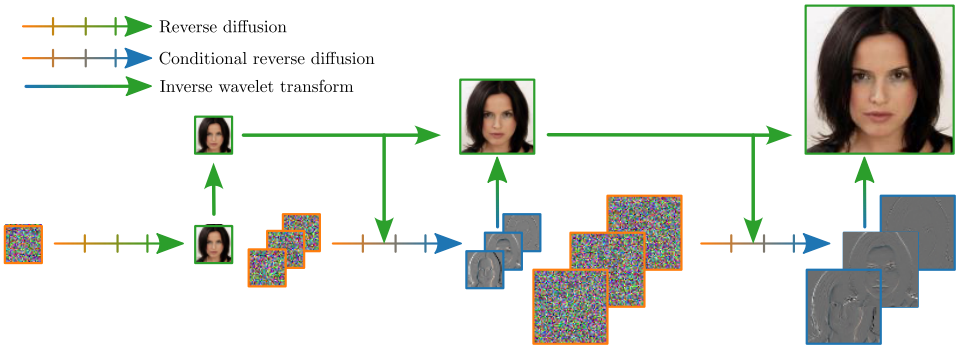

FIGURE 1.6: A wavelet score-based diffusion model combines score-based diffusion models (Figure 1.4) with a wavelet conditional factorization (Figure 1.5). Each factor in the decomposition of $p(x)$ is estimated with a dedicated score-based diffusion model. Conditional score-based diffusion models learn to denoise wavelet detail coefficients while exploiting the information contained in the clean image at the same resolution. Equivalently, this approach can be seen as a reparametrization of the time axis of score-based diffusion models, where iterations at large times $t \approx T$ are efficiently approximated with coarse-resolution images.

Fast generation with conditionally regular distributions. We show in Chapter 3 that the Lipschitz constant of the score controls the sampling complexity of score-based diffusion models. We explain that the scores of the conditional distributions $p(\bar{x}_j|x_j)$ have smaller Lipschitz constants than the global distribution $p(x)$. This is proved for stationary Gaussian distributions with a power spectrum that follows a power law. It is also demonstrated numerically for face images by showing that wavelet score-based diffusion models have a linear time complexity, as opposed to global score-based diffusion models of the entire image x. The informal reasoning is that a large conditioning number of the data covariance leads to irregular scores, and that covariance of image distributions are preconditioned in a wavelet basis. These results provide theoretical grounding for the use of multiscale approaches in score-based diffusion models.

Low-dimensional estimation with conditional Markov random fields. We then tackle in Chapter 4 the issue of score approximation. We prove that restricting the receptive field of the score network is equivalent to assuming a Markov random field model on the probability distribution $p(x)$, as well as the probability distributions of data contaminated with Gaussian white noise of any variance. We show that this Markov assumption is not satisfied by the global distribution $p(x)$, but is approximately satisfied by the wavelet conditional distributions $p(\bar{x}_j|x_j)$ in the case of face images. Additionally, the wavelet conditional distributions $p(\bar{x}_j|x_j)$ are stationary, in the sense that they are invariant to simultaneous translations of both $\bar{x}_j$ and x_j, as opposed to the global distribution $p(x)$ which is non-stationary. It thus leads to a multiscale stationary conditional Markov random field model, which alleviates the curse of dimensionality when learning score approximations.

Contributions. We demonstrate empirically that wavelet conditional factors of some natural image distributions are approximately stationary and local, and are amenable to faster sampling with score-based diffusion models. It is a first step in the study of the properties of the scores of natural image distributions, which might lead to a better understanding of the approximation and generalization properties of score networks.

1.3 Non-linear operators for image classification

In this section, we investigate the role of the non-linearity in deep convolutional neural networks trained on image classification. We consider two types of operators studied in previous works:

thresholdings in sparse representations, and phase collapses of wavelet coefficients. We investigate whether these mechanisms are relevant to understand the classification performance of deep convolutional networks. We show that phase collapses are both necessary and sufficient to reach high classification accuracies. This result allows defining more constrained approximation classes of the conditional energy $E(y|x)$ with structured architectures.

We begin by explaining in Section 1.3.1 how non-linearities may be classified according to their separation and concentration properties, of which soft-thresholding and phase collapses are characteristic examples. We then detail their respective properties in Sections 1.3.2 and 1.3.3.

1.3.1 Separation and concentration in deep networks

We review empirical results on the behavior of deep networks and introduce the distinction between separation and concentration operators.

Neural collapse. Image classification has been empirically solved by deep convolutional neural networks, with an accuracy that has improved with deeper networks (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016). Empirical observations have further shown that the accuracy of a linear probe classifier improves across layers (Zeiler and Fergus, 2014; Oyallon, 2017; Papyan, 2020), so that each layer contributes to the final objective. This calls for a study of the principles by which a non-linear layer can improve classification accuracy.

The accuracy of a linear classifier relies on the fact that input data points for different classes are confined to linearly separated regions. This can be achieved for instance by imposing that the means of each class are well separated, i.e., their distance is larger than the typical within-class variance. In fact, it has been observed that hidden representations at the last layer of deep networks may undergo a "neural collapse" (Papyan et al., 2020), where class means are maximally separated as the vertices of an equiangular tight frame, while within-class variance vanishes.

These empirical observations show that the non-linear operations in deep network layers progressively increase the linear separability of class means while concentrating within-class variance. What is the role of the network non-linearity in this phenomenon?

Separation and concentration. We introduce in Chapter 5 and refine in Chapter 6 an empirical distinction between two types of operators: those that separate class means on the one hand, and those that concentrate within-class variance on the other hand. This distinction is epitomized by even and odd non-linearities, of which every non-linearity is a linear combination. We shall in particular consider the absolute value $|\cdot|$ and the soft-thresholding $\rho_\lambda(t) = \mathrm{sgn}(t)\rho(|t| - \lambda)$ where ρ denotes the ReLU and $\lambda > 0$ is a positive threshold. These two non-linearities are archetypal, in the sense that the absolute value collapses the sign and preserves the amplitude, while applying a soft-thresholding preserves the sign and collapses small amplitudes. They roughly correspond to the even-odd decomposition of a ReLU with positive bias

$$\rho(t - \lambda) = \frac{1}{2}|\rho_\lambda(t)| + \frac{1}{2}\rho_\lambda(t), \tag{1.20}$$

as its even part is an absolute value (composed with a soft-thresholding) and its odd part is a soft-thresholding. A ReLU network may thus implement and rely on both of these non-linearities.

The following sections detail the properties of these two characteristic non-linearities and their role towards neural collapse. We explain in Section 1.3.2 that a soft-thresholding can concentrate "additive" within-class variability by leveraging a sparse decomposition of the class means. In contrast, we illustrate in Section 1.3.3 how an absolute value can separate class means by collapsing "multiplicative" within-class variability arising from a group. These two non-linearities can increase the Fisher ratio (Fisher, 1936; Rao, 1948), which measures the ratio of the

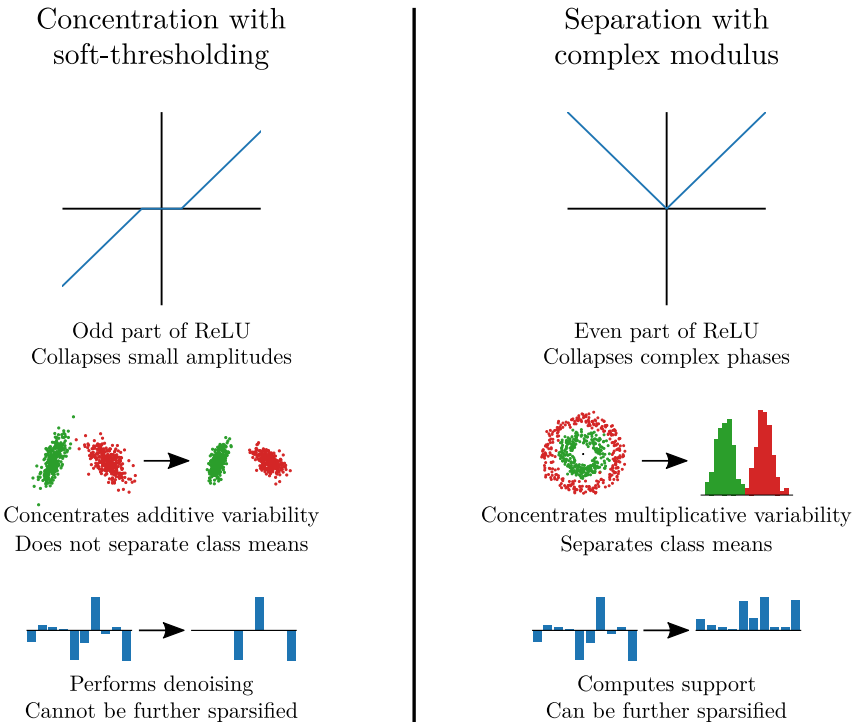

FIGURE 1.7: Comparison between soft-thresholding and absolute value/complex modulus non-linearities. Top: *behavior on scalar input.* Middle: *idealized behavior on points from two classes (green and red), visualized in two dimensions.* Bottom: *behavior on high-dimensional sparse vectors.*

class mean distance and the within-class variance, and thus both may increase the classification accuracy, though we will see that they are not equivalent. Their properties, detailed in the next two sections, are summarized in Figure 1.7.

1.3.2 Concentration with thresholdings in sparse representations

We study non-linear operators based on soft-thresholdings through a connection to sparse coding. We review the different flavors of sparse-coding operators, and highlight that a single soft-thresholding step already performs concentration of within-class variance under appropriate hypotheses.

Sparse coding. Consider a dictionary $D \in \mathbb{R}^{d \times m}$ composed of m "atoms" (elements) in $\mathbb{R}^d$. The ℓ^1 sparse coding problem (Tibshirani, 1996; Chen et al., 2001) consists in computing

$$z_{D,\lambda}(x) = \underset{z \in \mathbb{R}^m}{\arg\min} \ \frac{1}{2}\|x - Dz\|_2^2 + \lambda \|z\|_1, \tag{1.21}$$

for an input $x \in \mathbb{R}^d$, where $\lambda > 0$ is a threshold. The solution to this minimization problem can be computed with iterative soft-thresholding operations (Daubechies et al., 2004; Beck and Teboulle, 2009; Jiao et al., 2017). The number of iterations depends on the mutual coherence of the dictionary D, which characterizes its redundancy (Donoho and Elad, 2003). The dictionary D can be learned in a supervised way to maximize classification accuracy (Mairal et al., 2009, 2011) and can be incorporated in a network by unrolling the iterations of an optimization algorithm computing eq. (1.21) (Gregor and LeCun, 2010; Liu and Chen, 2019). In effect, the map $x \mapsto z_{D,\lambda}(x)$ is then a non-linear operator that can be used in a deep network. Depending on the use case, it comes in different flavors, which we now enumerate.

Embedding versus denoising. The non-linear operator $z_{D,\lambda}$ can separate close directions, as it can map correlated vectors to orthogonal vectors with non-overlapping supports. In contrast, the non-linear operator $D\, z_{D,\lambda}$ re-projects the sparse code to the input space and computes an approximation of the input x, as used in denoising by soft-thresholding (Donoho, 1995).

Explaining away versus bagging. The non-linear operator $z_{D,\lambda}$ can exploit the redundancy (coherence) of the dictionary D to perform a more complex operation. Redundant dictionaries require more iterations to compute the sparse code $z_{D,\lambda}(x)$, where the different atoms in D interact and compete to "explain away" the input x (Gregor and LeCun, 2010). In contrast, one can limit these computations to a single iteration, recovering a soft-thresholding

$$\hat{z}_{D,\lambda}(x) = \rho_\lambda(D^{\mathrm{T}}x). \tag{1.22}$$

This expression is equal to $z_{D,\lambda}(x)$ if D is orthogonal, but can also be used for more general dictionaries such as tight frames. The denoising version $D\rho_\lambda(D^{\mathrm{T}}x)$ can then be interpreted as a form of bagging by averaging denoised estimates computed in different orthogonal bases. The different levels of redundancy of the dictionary and their associated complexity of computing $z_{D,\lambda}$ are characteristic of the evolution of sparse representations in signal processing, from wavelet orthogonal bases to the curvelet tight frame to the bandlet best-basis search algorithm.

Non-negative sparse codes. A final variant on the sparse-coding non-linear operator is that eq. (1.21) can be modified to compute a non-negative sparse code

$$z^+_{D,\lambda}(x) = \underset{z\in\mathbb{R}^m_+}{\arg\min}\ \frac{1}{2}\|x - Dz\|_2^2 + \lambda\|z\|_1. \tag{1.23}$$

This variant has been used in several works that interpret iterations on ReLUs and linear operators in deep networks as computing *non-negative* sparse codes (Sun et al., 2018; Sulam et al., 2018, 2019; Mahdizadehaghdam et al., 2019; Zarka et al., 2020). The solution of eq. (1.23) can be computed similarly to the solution of eq. (1.21) by replacing soft-thresholdings $\rho_\lambda(t)$ by biased ReLUs $\rho(t-\lambda)$ in the iterations of the optimization algorithm. However, as announced in Section 1.3.1 and will be detailed in Section 1.3.3, this breaking of the sign symmetry confers very different properties to the non-linear operator $z^+_{D,\lambda}$. We shall therefore focus on soft-thresholding-based sparse coding in this dissertation.

Prior work on sparse coding in deep networks. We have presented several non-linear operators based on soft-thresholdings. Which properties of these sparse-coding non-linearities are useful for classification and what is their role in producing a neural collapse? This line of research was started with a numerical investigation in the prior work of Zarka et al. (2020). The authors show that applying the non-linear operator $z^+_{D,\lambda}$ on top of the scattering transform (Mallat, 2012; Bruna and Mallat, 2013) followed by a multi-layer perceptron classifier allows reaching the classification performance of AlexNet (Krizhevsky et al., 2012) on the ImageNet dataset (Russakovsky et al., 2015). Additionally, $z^+_{D,\lambda}$ can be replaced with $D\, z^+_{D,\lambda}$ at a negligible cost in accuracy. Further experiments done by the authors (communicated in personal correspondence, but some are reported in Zarka (2022, Chapter 3)) show that the non-negativity constraint can be dropped, and that a single thresholding iteration is enough to capture most of the accuracy improvements. The sparse-coding operation of eq. (1.21) is thus reduced to the operator $D\rho_\lambda D^{\mathrm{T}}$ in the setting of Zarka et al. (2020). It thus remains to assess the generality of these findings (do they hold in other architectures with higher accuracies?) and understand the properties of the operator $D\rho_\lambda D^{\mathrm{T}}$ for classification.

In this work. In order to understand the properties of thresholdings for classification, we introduce in Chapter 5 a stylized clutter model, which assumes that each class is distributed as Gaussian mixture. If the means of each mixture component are efficiently approximated in an orthogonal dictionary D, we prove that the non-linear soft-thresholding operator $D\rho_\lambda D^{\mathrm{T}}$ concentrates the variance of each mixture component while preserving the separation between their means. This theorem provides a justification for the use of sparsity-inducing non-linearities in image classification, without the need for explaining away between dictionary atoms. It shows that a soft-thresholding reduces "additive" within-class variability by leveraging a sparse representation of separated class means. The operator $D\rho_\lambda D^{\mathrm{T}}$ can then be used in a deep network in conjunction with other operators that separate class means. This is validated numerically in the learned scattering architecture of Chapter 5, which reaches the accuracy of ResNet-18 (He et al., 2016) on ImageNet. The properties of $D\rho_\lambda D^{\mathrm{T}}$ are further examined and contrasted with phase collapse operators in the next section.

1.3.3 Separation with phase collapses of wavelet coefficients

The previous section explained how a soft-thresholding can concentrate "additive" within-class variability. It however requires the class means to be separated. We explain that this is usually not the case due to geometric within-class variability, which creates stationary phases and collapses all class means to zero. This "multiplicative" within-class variability is concentrated with a complex modulus, which can separate class means. We detail this mechanism and compare its properties with soft-thresholdings as a non-linearity in image classification.

Translation variability and stationary phases. Image classes are typically stationary, in the sense that the distribution of x is invariant to translations when conditioned on a given class y. It means that translations give rise to within-class variability. This translation variability is best understood in the Fourier domain, where it leads to stationary complex phases. This implies that class means are zero for non-zero frequencies, as $\hat{x}(\omega)$ has a circularly-symmetric distribution when conditioned on y for a frequency $\omega \neq 0$. Group variability is therefore of multiplicative nature due to its representation as phase shifts. This multiplicative variability prevents class means from being separated. It follows that to improve linear classification accuracy, the non-linearity *must* (at least partially) collapse these stationary phases, which may separate class means.

We have focused on translations for simplicity of exposition, but this discussion can be generalized to other groups by considering generalized Fourier transforms defined from irreducible representations of the group (see, e.g., Cohen et al., 2018; Kondor et al., 2018, for rotations in $\mathbb{R}^3$).

Phase collapses and the scattering transform. The phase collapse phenomenon is used iteratively by the scattering transform (Mallat, 2012). It considers variability arising from the deformations rather than translations, and thus replaces the Fourier transform with the wavelet transform. A deformation of the input image is approximately decomposed with a wavelet filter as a spatially-varying phase and a larger-scale deformation. A complex modulus then eliminates this phase, thus creating approximate invariants to small-scale deformations. Invariants to larger-scale deformations are obtained by iterating this process with another set of wavelet filters and complex moduli. The scattering transform thus realizes a "non-linear hierarchical diagonalization" of deformations, which are represented by a cascade of spatially-varying phases at all scales before being collapsed.

Comparing thresholdings and phase collapses. The distinction between even and odd non-linearities made in Section 1.3.1 can be generalized in the complex domain to decompositions

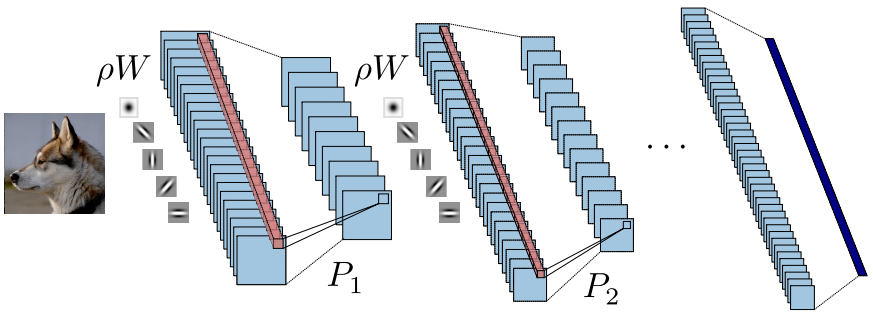

FIGURE 1.8: The learned scattering architecture introduced in Chapter 6. It iterates phase collapses ρW, where W is a spatial convolution with predefined wavelet filters and $\rho(z) = (|z|, z)$ is a modulus with skip-connection, and learned projectors $(P_j)_{j \leq J}$ that apply over channels only.

into phase harmonics (Mallat et al., 2019). Phase-harmonic non-linearities of order $k \in \mathbb{Z}$ are such that $\rho(\mathrm{e}^{i\alpha} z) = \mathrm{e}^{ik\alpha}\rho(z)$, and any complex non-linearity can be decomposed uniquely as a sum of phase-harmonic non-linearities of all orders. We will consider only phase harmonics of order 0 and 1, as higher-order phase harmonics accelerate the phase and are thus unstable when iterated, while phase harmonics with negative orders are redundant as they can be obtained by complex conjugation. Compared to the discussion in Section 1.3.1, the sign is then replaced by the complex phase, the absolute value by the complex modulus, and the soft-thresholding by a complex soft-thresholding $\rho_\lambda(z) = \rho(|z| - \lambda)\mathrm{e}^{i\varphi(z)}$ where $\varphi(z)$ is the phase of $z \in \mathbb{C}$.

What is the relationship between these two qualitatively opposite non-linear mechanisms? On the one hand, the modulus is a phase harmonic of order 0, which collapses the phase and preserves the amplitude. It separates class means and concentrates multiplicative within-class variability by diagonalizing a group action. On the other hand, the soft-thresholding is a phase harmonic of order 1, which preserves the phase and collapses small amplitudes. It concentrates additive within-class variability by leveraging a sparse representation of separated class means. A common point is that in the case of images, both non-linearities should be computed in the wavelet domain, leveraging respectively its diagonalization and sparsity properties. Indeed, it can be expected more generally that diagonalizing geometric variability is required to obtain a sparse representation, in order to avoid that a small deformation dilutes the large coefficients and greatly increases the support size. However, a major difference between these two mechanisms is that they have different behaviors when iterated.

Iterating non-linearities. We show in Chapter 6 that a sparse representation necessarily concentrates the entropy of the process in the phases of the coefficients. If the non-linearity preserves these phases, such as a soft-thresholding, then the process cannot be further "sparsified". It implies that applying a second soft-thresholding in a learned dictionary would not lead to accuracy improvements, as observed numerically by Zarka (2022). This maximal-entropy property is consistent with the observations that the phases of wavelet coefficients tend to be stationary and are approximately independent (Wainwright and Simoncelli, 1999).

In contrast, iterating phase collapses leads to improved classification accuracies, as evidenced by the scattering transform (Bruna and Mallat, 2013). Phase collapses eliminate the entropy contained in the phases, and allow extracting linearly the support of large amplitude coefficients. The geometric regularity of this support set across space, scale, and orientation can then be exploited by the filters at the next layer to obtain an even sparser representation. It shows that iterating sparsity *requires* collapsing the phases of the intermediate sparse codes. The properties of soft-thresholdings and phase collapses are summarized in Figure 1.7.

Numerical experiments. We define in Chapters 5 and 6 learned scattering network architectures whose non-linearities can be attributed to only one of the two opposite mechanisms

presented above. The architectures leverage prior information by using fixed wavelet filters across space and learning only filters across channels. They reach the same accuracy as ResNet-18 on the ImageNet dataset despite having fewer layers.

In addition to the theoretical considerations above, we compare experimentally the two mechanisms in Chapter 6 and demonstrate that phase collapses are both necessary and sufficient to reach high classification accuracies, while soft-thresholdings or any other phase harmonic of order 1 are neither necessary nor sufficient. This is demonstrated on both learned scattering networks, with complex wavelet filters, and ResNet-18, with real-valued learned filters.

Contributions. Our results show that the principal non-linearity mechanism used by deep networks to increase linear classification accuracy is the iteration of phase collapses of wavelet coefficients. The resulting learned scattering network, illustrated in Figure 1.8, can then be seen as a structured architecture with a maximum amount of components defined from prior information and minimal learning.

1.4 A model of network weights with aligned random features

The previous section has shown that prior information can be leveraged to define non-linear functional blocks and structure deep network architectures. However, it remains to learn weight matrices acting along channels. We thus turn to the study the learned weights in deep networks in order to understand their mathematical structure and the properties of the associated learned representations.

This has been the focus of a series of prior works that aimed at understanding the performance gap between the scattering transform and deep convolutional networks. Following the early successes of methods based on group invariants, a first line of research devised extensions of the scattering transform to build invariants to larger groups, incorporating rotations and scalings (Sifre and Mallat, 2013; Oyallon and Mallat, 2015) or frequency modulations in audio (Andén et al., 2015). A second line of work then studied empirically the learned operators (Oyallon et al., 2017; Oyallon, 2017), trying to characterize them in terms of computing learned invariants to unknown symmetry groups, as suggested by Mallat (2016).

In this dissertation, we adopt a different direction, and rather follow the random-feature kernel viewpoint developed in the literature. We begin by presenting this viewpoint in Section 1.4.1. We then enumerate several attempts at extending it to trained networks in Section 1.4.2. We introduce our own contributions in Section 1.4.3, which are based on the idea of random feature alignment. They are the key ingredient in the proposed probabilistic model of trained weights.

1.4.1 Random-feature kernels in deep networks

We review in this section the convergence properties of random-feature networks and their associated hierarchical kernels.

Random-feature networks. Networks are typically initialized with random weights. At initialization, deep networks thus compute random hidden representations $\hat{\phi}_j(x) \in \mathbb{R}^{d_j}$ defined iteratively at each layer j by

$$\hat{\phi}_j(x) = \Big(\rho(\langle w_{j,i}, \hat{\phi}_{j-1}(x)\rangle)\Big)_{i \leq d_j} \quad \text{with i.i.d. } w_{j,i} \sim \mathcal{N}(0, \mathrm{Id}), \tag{1.24}$$

starting from $\hat{\phi}_0(x) = x$, where ρ is a pointwise non-linearity such as ReLU. Such networks compute *random features* (Rahimi and Recht, 2007). Random-feature networks can already achieve a performance comparable to trained networks on simple tasks, provided that the architecture (non-linearities, pooling operations, etc) is appropriately designed (Jarrett et al., 2009; Pinto et al., 2009).

Random-feature kernels. The random feature maps $\hat{\phi}_j$ define random kernels $\hat{k}_j$ at each layer,[2]

$$\hat{k}_j(x, x') = \langle \hat{\phi}_j(x), \hat{\phi}_j(x') \rangle = \frac{1}{d_j} \sum_{i=1}^{d_j} \rho(\langle w_{j,i}, \hat{\phi}_{j-1}(x) \rangle) \, \rho(\langle w_{j,i}, \hat{\phi}_{j-1}(x') \rangle). \tag{1.25}$$

These kernels depend on the particular realization of the random features $(w_{j,i})_{j,i}$, but when the widths $(d_j)_j$ increase to infinity they converge to a deterministic kernel (Rahimi and Recht, 2007; Daniely et al., 2016), defined by

$$k_j(x, x') = \langle \phi_j(x), \phi_j(x') \rangle = \mathbb{E}_{w_j \sim \mathcal{N}(0, \mathrm{Id})} \Big[\rho(\langle w_j, \phi_{j-1}(x) \rangle) \, \rho(\langle w_j, \phi_{j-1}(x') \rangle) \Big], \tag{1.26}$$

where ϕ_j is an associated kernel embedding or feature map (Aronszajn, 1950). For Gaussian white initializations $w_{j,i} \sim \mathcal{N}(0, \mathrm{Id})$, this kernel is a dot-product kernel and admits a closed-form expression for several non-linearities ρ including ReLU (Cho and Saul, 2009; Daniely et al., 2016).

These kernels are sometimes referred to as "conjugate" kernels (Fan and Wang, 2020), by opposition to the neural tangent kernel. They also arise in the equivalence between random neural networks and Gaussian processes. Indeed, the random projections $\langle w_{j+1,i}, \hat{\phi}_j(x) \rangle$ converge in the infinite-width limit to a zero-mean Gaussian process whose covariance kernel is given by eq. (1.26) (Neal, 1996; Williams, 1996; Lee et al., 2018; Matthews et al., 2018).

The analysis of random-feature approximations (Rahimi and Recht, 2008; Bach, 2017b; Rudi and Rosasco, 2017; Mei et al., 2022; Schröder et al., 2023) allows linking the behavior of finite-width networks at initialization to properties of the deterministic kernels in eq. (1.26). Computing their spectrum (Scetbon and Harchaoui, 2021) then allows studying effects of the architecture such as the benefits (or lack thereof) of depth (Bietti and Bach, 2021). The behavior of the asymptotic kernels was also studied in the signal propagation literature (starting with Poole et al., 2016; Schoenholz et al., 2017), leading to measures of trainability of the network at initialization.

Hierarchical kernel models. Several works in the literature have introduced hierarchical kernel models inspired from neural network architectures with random features, following the work of Cho and Saul (2009). Mairal et al. (2014) introduce kernels obtained from the composition of patch extractions, dot-product kernel embeddings, and pooling operators. The stability to deformations properties of these kernels are studied in Bietti and Mairal (2019). One can then incorporate projections in learned subspaces in-between the kernel embeddings (Cho and Saul, 2009; Mairal, 2016). A correspondence can be established between convolutional architectures and such hierarchical kernels (Anselmi et al., 2015).

1.4.2 Evolution of kernels and training dynamics

While the kernel viewpoint has been fruitful to describe networks at initialization, it is not clear to what extent trained networks may be described by a (possibly data-dependent) kernel. We now describe several prior attempts towards describing trained networks, and relevant numerical observations in the literature.

The neural tangent kernel in the lazy regime. Depending on the scalings of the network initialization, renormalization factors, and learning rate (Yang and Hu, 2021), the behavior of some trained networks remains described by a different kernel (the neural tangent kernel, Jacot et al., 2018; Lee et al., 2019b), which becomes deterministic and fixed during training in the considered infinite-width limit. This is the so-called lazy regime (Chizat et al., 2019), which

[2] We sometimes add or omit normalizing factors such as d_j^{-1} to lighten the notations.

does not account for the performance of trained deep networks on complex tasks (Lee et al., 2020; Geiger et al., 2020).

Evolution of empirical kernels in the feature-learning regime. The lazy regime has been opposed to a rich or feature-learning regime, in which the conjugate and tangent kernels evolve during training. Several works have studied the evolution of empirical conjugate kernels (Fischer et al., 2022; Seroussi et al., 2023) or empirical tangent kernels (Shan and Bordelon, 2021; Atanasov et al., 2022). They report an alignment period in which the eigenvectors of the kernel move rapidly, followed by an amplification along the first kernel principal components. This is consistent with numerical experiments (Fort et al., 2020; Baratin et al., 2021). A critical empirical observation made by Raghu et al. (2017); Kornblith et al. (2019) is that the empirical conjugate kernels learned by two independently trained networks are increasingly similar when the network width increases, suggesting a convergence to a deterministic kernel in the infinite-width limit.

Mean-field limit. The feature-learning regime is however more challenging to study and has thus only been characterized in simplifying cases. In particular, one-hidden-layer networks have been studied in the *mean-field* infinite-width limit (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020). The network is characterized with the empirical neuron weight distribution of the first layer

$$\hat{\pi}_1 = \frac{1}{d_1} \sum_{i=1}^{d_1} \delta_{w_{1,i}}. \tag{1.27}$$

$\hat{\pi}_1$ converges in the infinite-width limit to a deterministic measure π_1 which evolves during training as a Wasserstein gradient flow. The network can thus be described with a time-dependent random-feature kernel as in eq. (1.26) whose weight distribution changes with time. The limit distribution at the end of training can then be characterized as minimizing a total-variation norm (Chizat and Bach, 2020).

Statistical properties of trained weights. Despite several attempts, the generalization of the mean-field limit to deeper networks remains elusive (Sirignano and Spiliopoulos, 2022; E and Wojtowytsch, 2020; Nguyen and Pham, 2020; Chen et al., 2022c; Yang and Hu, 2021). Several papers have thus resorted to empirical studies of the weight statistics of trained networks, especially through their singular value distributions (Martin and Mahoney, 2021; Thamm et al., 2022). These measurements recover the observation that deep networks learn low-rank weight matrices (Denil et al., 2013; Denton et al., 2014; Yu et al., 2017).

1.4.3 Alignment convergence: the rainbow model

The works reviewed in Sections 1.4.1 and 1.4.2 show that the study of neural networks through the prism of their associated kernels is very fruitful. In particular, the law of large numbers plays a central role by implying a convergence of the conjugate and tangent random-feature kernels . The optimization and generalization properties of deep networks can then be linked to properties of these kernels. Theoretical analyses have been restricted to either untrained or shallow networks, but empirical observations suggest that this viewpoint may be generalized.

We now present an extension of this viewpoint to trained deep networks which is detailed in Chapter 7. The central ingredient is the empirical observation (following Kornblith et al. (2019)) that hidden activations $\hat{\phi}_j$ converge up to a rotation, and that *this rotation can be absorbed by the next layer weights*. This observation motivates a conjecture for a static mean-field limit of deep networks. It then leads to a probabilistic model of network weights.

A conjectured multi-layer static mean-field limit. We now informally introduce a conjecture for a multi-layer generalization of the mean-field limit, in order to motivate the model presented in the next paragraph. This conjecture is inspired by the idea that that deep networks have an internal rotation symmetry at each layer, but are otherwise entirely determined up to this symmetry, as will be confirmed in our numerical experiments. More precisely, the conjecture states that after an alignment procedure, both activations and weight distributions converge in the infinite-width limit. The arbitrary rotations describing the trained network arise from the stochasticity of the training process (coming from the random initialization, the batch ordering in SGD, and the data augmentation).

The conjectures assumes that for each layer j, there exists a deterministic feature map ϕ_j defined in a separable Hilbert space H_j and a partial isometry $A_j \colon \mathbb{R}^{d_j} \to H_j$ that depends on $\hat{\phi}_j$ such that

$$A_j \, \hat{\phi}_j \to \phi_j, \tag{1.28}$$

in mean square distance when the widths of the network increase to infinity. The partial isometry A_j computes an alignment of $\hat{\phi}_j$ to ϕ_j and is defined by minimizing the mean square error

$$\min_{A_j^{\mathrm{T}} A_j = \mathrm{Id}_{d_j}} \mathbb{E}_x \Big[\| A_j \, \hat{\phi}_j - \phi_j \|_{H_j}^2 \Big]. \tag{1.29}$$

For convenience, we also define $\hat{\phi}_0(x) = \phi_0(x) = x$ and $A_0 = \mathrm{Id}$ with $H_0 = \mathbb{R}^{d_0}$.

The alignment matrices are used to define the aligned layer operations $A_j \, \rho W_j \, A_{j-1}^{\mathrm{T}}$, which iteratively compute the aligned activations $A_j \, \hat{\phi}_j(x)$. We thus define the aligned empirical weight distributions $\hat{\pi}_j$ defined from the aligned weight matrices $W_j A_{j-1}^{\mathrm{T}}$:

$$\hat{\pi}_j = \frac{1}{d_j} \sum_{i=1}^{d_j} \delta_{A_{j-1} w_{j,i}}, \tag{1.30}$$

where the $(w_{j,i})_{i \leq d_j}$ are the d_j rows of W_j. The conjecture assumes that for each layer j, $\hat{\pi}_j$ converges to a distribution π_j defined on H_{j-1}:

$$\hat{\pi}_j \to \pi_j, \tag{1.31}$$

in mean-square Wasserstein-2 distance when the layer widths increase to infinity.

The two statements of the conjecture, namely eqs. (1.28) and (1.31), are not independent. We sketch an informal reasoning which derives by induction the convergence of aligned activations and weight distributions at all layers. The induction is initialized with $A_0 \hat{\phi}_0 \to \phi_0$ which holds by definition. Assume that $A_j \hat{\phi}_{j-1} \to \phi_{j-1}$. By analogy with the mean-field limit of one-hidden-layer-networks, it seems natural that it implies a convergence on the aligned weight distributions at the next layer $\hat{\pi}_j \to \pi_j$. The convergences of $A_{j-1} \hat{\phi}_j$ and $\hat{\pi}_j$ then impose a convergence on the kernel defined by $\hat{\phi}_j$:

$$\Big\langle \hat{\phi}_j(x), \hat{\phi}_j(x') \Big\rangle = \frac{1}{d_j} \sum_{i=1}^{d_j} \rho\Big(\Big\langle A_{j-1} w_{j,i}, A_{j-1} \hat{\phi}_{j-1}(x) \Big\rangle \Big) \, \rho\Big(\Big\langle A_{j-1} w_{j,i}, A_{j-1} \hat{\phi}_{j-1}(x') \Big\rangle \Big) \tag{1.32}$$

$$\to \mathbb{E}_{w_j \sim \pi_j} \Big[\rho(\langle w_j, \phi_{j-1}(x) \rangle) \, \rho\Big(\langle w_j, \phi_{j-1}(x') \rangle \Big) \Big] = \Big\langle \phi_j(x), \phi_j(x') \Big\rangle. \tag{1.33}$$

This defines the feature map ϕ_j (up to rotation) from ϕ_{j-1} and π_j, and implies that the kernel of $\hat{\phi}_j$ converges to the kernel of ϕ_j. One can then prove that it implies that $A_j \, \hat{\phi}_j \to \phi_j$, completing the induction step.

We verify numerically in Chapter 7 the convergence in eq. (1.28) of aligned activations on learned scattering networks and ResNets trained on the CIFAR-10 and ImageNet datasets. We also verify the convergence in of eq. (1.31) of the aligned weight distributions on learned scattering networks trained on CIFAR-10 through the convergence of their covariance. These results provide empirical evidence for the mean-field conjecture.

The rainbow model. Chapter 7 introduces the rainbow model of deep networks. It is a model which specifies the joint probability distribution of trained network weights across layers, and is motivated by the reasoning in the above paragraph. The model is parameterized by weight distributions $(\pi_j)_{j\leq J}$ and activations $(\phi_j)_{j\leq J}$ which satisfy the consistency equation (1.33). It assumes that the neurons $w_{j,i}$ are independent samples from π_j that have been aligned to the activations of the previous layer with A_{j-1}:

$$w_{j,i} = A_{j-1}^{\mathrm{T}} w'_{j,i} \quad \text{with } w'_{j,i} \sim \pi_j \text{ independently.} \tag{1.34}$$

The model can be sampled iteratively as follows: the first layer weights $(w_{1,i})_{i\leq d_1}$ are i.i.d. samples from π_1. They define the activations $\hat{\phi}_1$, from which the alignment A_1 to ϕ_1 can be obtained. The weights at the second layer $(w_{2,i})_{i\leq d_2}$ are then i.i.d. samples from the finite-dimensional marginal of π_2 given by A_1^{T}. It similarly defines the activations $\hat{\pi}_2$ and the alignment A_2, and the process is repeated for all layers. The different layer weights are thus not independent: W_j depends on the previous layer weights $W_1, \dots, W_{j-1}$ through the alignment A_{j-1}.

Under this simplified probabilistic model, we prove in Chapter 7 that we recover the activation convergence of eq. (1.28). The argument relies on the law of large numbers applied to the kernels as in eqs. (1.32) and (1.33). Even though the rainbow assumptions may be too restrictive to apply to trained weights, they apply at initialization with $\pi_j = \mathcal{N}(0, \mathrm{Id})$. This result thus may be of interest to the study of the SGD training dynamics.

In practice, one needs to estimate finite-dimensional approximations of the infinite-dimensional feature maps $(\phi_j)_{j\leq J}$ and weight distributions $(\pi_j)_{j\leq J}$. In our numerical experiments, we approximate ϕ_j the activations of a large but finite-width network. It then remains to define parameterized models of the weight distributions π_j.

Colored weight covariances. Rainbow networks, and any network which shares the convergence properties stated in eqs. (1.28) and (1.31), thus implement a deterministic function which is in a reproducing kernel Hilbert space (RKHS). The properties of this RKHS are determined by the weight distributions $(\pi_j)_{j\leq J}$. In particular, the singular values of the weight matrices W_j, or equivalently the eigenvalues of the empirical aligned weight covariance matrices $d_j^{-1} A_{j-1} W_j^{\mathrm{T}} W_j A_{j-1}^{\mathrm{T}}$, are related to the eigenvalues of the weight covariance matrices

$$C_j = \mathbb{E}_{w_j \sim \pi_j}\left[w_j w_j^{\mathrm{T}}\right]. \tag{1.35}$$

The covariances C_j thus capture the reductions in dimensionality computed by the weight matrices W_j.

The effect of the covariances can be evidenced by factorizing

$$W_j = \tilde{W}_j \, \hat{C}_j^{1/2}, \tag{1.36}$$

where $\hat{C}_j = A_{j-1}^{\mathrm{T}} C_j A_{j-1}$ is the covariance expressed in the basis defined by the activations $\hat{\phi}_{j-1}$, and $\tilde{W}_j$ are the whitened weights which thus have an identity covariance.

Rainbow networks thus iterates between the linear dimensionality reductions computed by the "colored" covariances $\hat{C}_j^{1/2}$ and non-linear high-dimensional embeddings with the white random features $\rho \tilde{W}_j$. This is illustrated in Figure 1.9. Similar models based on hierarchical kernels were introduced in previous works (Cho and Saul, 2009; Anselmi et al., 2015; Mairal, 2016; Bietti, 2019).

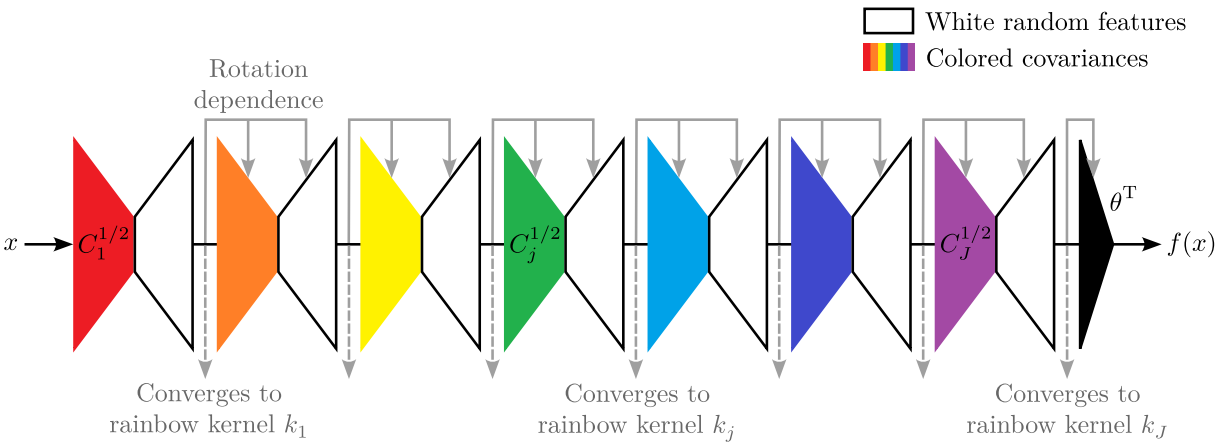

FIGURE 1.9: The rainbow model models network operations as an alternation of increases and decreases in dimensionality. The former is due to the low-rank weight covariances, while the latter is due to the non-linear white random features. The random features give rise to arbitrary rotations in the activations at each layer, which in turn create dependencies with the weights at the next layer which are similarly rotated. After the alignment procedure, both activations and weight distributions converge to a deterministic limit.

Gaussian rainbow networks. In Chapter 7, we also specialize the rainbow model to the Gaussian case, where $\pi_j = \mathcal{N}(0, C_j)$ is entirely determined by its covariance. We show that Gaussian rainbow networks enjoy several theoretical properties due to the rotation invariance of the multivariate normal distribution. First, the white random features $\rho\tilde{W}_j$ then compute dot-product kernel embeddings. Second, it also leads to approximate equivariance with respect to subgroups of the orthogonal group, and it becomes exact when the widths increase to infinity.

The Gaussian assumption is too restrictive to model arbitrary trained networks. However, it can approximately hold for architectures which incorporate prior information and restrict their learned weights. We show in Chapter 7 that the learned scattering network architecture described in Chapter 6 learns approximately Gaussian channel weights. In particular, sampling conditionally Gaussian weights according to eq. (1.34) using weight covariances estimated from the weights of a trained network leads to a comparable classification accuracy without training when the network width is large enough.

Contributions. The rainbow model is a joint model of the probability distribution of trained network weights. It integrates empirical observations on the statistical properties of network weights and activations. The alignment procedure performs a "registration" or "canonicalization" of the network by "fixing the gauge symmetry" of rotations in the network hidden layers, which greatly facilitates numerical and theoretical analyses. The rainbow model may lead to new insights in the approximation and generalization properties of deep networks. In addition, the conjectured static mean-field limit, which holds at initialization, might enable refined studies of the training dynamics of stochastic gradient descent in deep networks.

1.5 Organization of the dissertation

The work presented in this dissertation has resulted in five conference papers and one preprint:

1. Florentin Guth*, Etienne Lempereur*, Joan Bruna, and Stéphane Mallat. Conditionally strongly log-concave generative models. In *International Conference on Machine Learning*, 2023.

2. Florentin Guth, Simon Coste, Valentin De Bortoli, and Stéphane Mallat. Wavelet score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2022.

3. Zahra Kadkhodaie, Florentin Guth, Stéphane Mallat, and Eero P Simoncelli. Learning multi-scale local conditional probability models of images. In *International Conference on Learning Representations*, 2023.

4. John Zarka, Florentin Guth, and Stéphane Mallat. Separation and concentration in deep networks. In *International Conference on Learning Representations*, 2021.

5. Florentin Guth, John Zarka, and Stéphane Mallat. Phase collapse in neural networks. In *International Conference on Learning Representations*, 2022.

6. Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *arXiv preprint arXiv:2305.18512*, 2023.

They are collected in the rest of the manuscript, which is divided in three parts.

In Part I, we study the properties of the wavelet conditional distributions introduced by Marchand et al. (2022). We generalize the wavelet conditional factorization to wavelet packet projectors in Chapter 2. It leads to conditionally log-concave models of multiscale physical fields, for which all sources of errors can be controlled. We then turn to score-based diffusion models in Chapter 3. We show that the cascaded diffusion models introduced by Saharia et al. (2021); Ho et al. (2022); Dhariwal and Nichol (2021) leverage the regularity of the conditional wavelet scores to accelerate the generation of samples from the model. We then focus in Chapter 4 on estimation of the scores with deep networks. We show that the wavelet conditional distributions of face images are well approximated with a local Markov random field, which allows reducing the receptive field size of the score network and hence the dimensionality of the learning task.

In Part II, we study structured non-linear operators for image classification. We introduce in Chapter 5 distinct separation and concentration operators for image classification. We demonstrate that these non-linear operators can increase the linear classification accuracy via different means. These operators are combined in a first learned scattering network architecture which reaches the classification accuracy of ResNet-18 on the ImageNet dataset. In Chapter 6, we then specialize the separation operator of the previous chapter to the phase collapse of complex wavelet coefficients. We show mathematically and numerically that these phase collapses are both necessary and sufficient to obtain high classification accuracies. This leads to a second, more streamlined learned scattering network architecture which exploits exclusively this non-linear mechanism to increase classification accuracy.

In Part III, we study the weights of trained deep networks. We introduce in Chapter 7 a model of their probability distribution. We prove that the assumptions of the model imply that aligned hidden activations converge in the infinite-width limit, recovering the observations of Kornblith et al. (2019). The various model properties are validated numerically on ResNets or learned scattering networks. In particular, we demonstrate that Gaussian rainbow networks provide accurate models of learned scattering networks trained on the CIFAR-10 dataset.

Finally, we conclude in Chapter 8 with a summary of our findings and perspectives for future work.

Part I

Properties of Wavelet Conditional Probability Distributions

Chapter

2 Conditionally Strongly Log-Concave Generative Models

Chapter content

2.1 Introduction . . . 32
2.2 Conditionally strongly log-concave models . . . 33
2.2.1 Conditional factorization and log-concavity . . . 33
2.2.2 Learning guarantees with score matching . . . 35
2.2.3 Score matching with exponential families . . . 36
2.2.4 Sampling guarantees with MALA . . . 37
2.3 Wavelet packet conditional log-concavity . . . 38
2.3.1 Energies with scalar potentials . . . 38
2.3.2 Wavelet packets and renormalization group . . . 39
2.3.3 Multiscale scalar potentials . . . 39
2.4 Numerical results . . . 40
2.4.1 φ^4 scalar potential energy . . . 40
2.4.2 Conditional log-concavity . . . 41
2.4.3 Application to cosmological data . . . 43
2.5 Discussion . . . 44

There is a growing gap between the impressive results of deep image generative models and classical algorithms that offer theoretical guarantees. The former suffer from mode collapse or memorization issues, limiting their application to scientific data. The latter require restrictive assumptions such as log-concavity to escape the curse of dimensionality. We partially bridge this gap by introducing conditionally strongly log-concave (CSLC) models, which factorize the data distribution into a product of conditional probability distributions that are strongly log-concave. This factorization is obtained with orthogonal projectors adapted to the data distribution. It leads to efficient parameter estimation and sampling algorithms, with theoretical guarantees, although the data distribution is not globally log-concave. We show that several challenging multiscale processes are conditionally log-concave using wavelet packet orthogonal projectors. Numerical results are shown for physical fields such as the φ^4 model and weak lensing convergence maps with higher resolution than in previous works. These results evidence properties of some image distributions that may be used to escape the curse of dimensionality.

This chapter is adapted from the following publication: Florentin Guth*, Etienne Lempereur*, Joan Bruna, and Stéphane Mallat. Conditionally strongly log-concave generative models. In *International Conference on Machine Learning*, 2023. We omit the proofs of the mathematical results in Section 2.2, which were not done by the author of this dissertation.

2.1 Introduction

Generative modeling requires the ability to estimate an accurate model of a probability distribution from a training dataset, as well as the ability to efficiently sample from this model. Any such procedure necessarily introduces errors, due to limited expressivity of the model class, learning errors of selecting the best model within that class, and sampling errors due to limited computational resources. For high-dimensional data, it is highly challenging to control all errors with polynomial-time algorithms. Overcoming the curse of dimensionality requires exploiting structural properties of the probability distribution. For instance, theoretical guarantees can be obtained with restrictive assumptions of log-concavity, or with low-dimensional parameterized models. In contrast, recent deep-learning-based approaches such as diffusion models (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022) have obtained impressive results for distributions which do not satisfy these assumptions. Unfortunately, in such cases, theoretical guarantees are lacking, and diffusion models have been found to memorize their training data (Carlini et al., 2023; Somepalli et al., 2022), which is inappropriate for scientific applications. The disparity between these two approaches highlights the need for models which combine theoretical guarantees with sufficient expressive power. This chapter contributes to this objective by defining the class of conditionally strongly log-concave distributions. We show that it is sufficiently rich to model the probability distributions of complex multiscale physical fields, and that such models can be sampled with fast algorithms with provable guarantees.

Sampling and learning guarantees. While the theory for sampling log-concave distributions is well-developed (Chewi, 2023), simultaneous learning and sampling guarantees for general non-log-concave distributions are less common. Block et al. (2020) establish a fast mixing rate of multiscale Langevin dynamics under a manifold hypothesis. Koehler et al. (2022) studies the asymptotic efficiency of score-matching compared to maximum-likelihood estimation under a global log-Sobolev inequality, which is not quantitative beyond globally log-concave distributions. Chen et al. (2022b,a) establish polynomial sampling guarantees for a reverse score-based diffusion, given a sufficiently accurate estimate of the time-dependent score. Sriperumbudur et al. (2013); Sutherland et al. (2018); Domingo-Enrich et al. (2021) study density estimation with energy-based models under different infinite-dimensional parametrizations of the energy. They use various metrics including score-matching to establish statistical guarantees that avoid the curse of dimensionality, under strong smoothness or sparsity assumptions of the target distribution. Finally, Balasubramanian et al. (2022) derive sampling guarantees in Fisher divergence of Langevin Monte-Carlo beyond log-concave distributions. While these hold under a general class of target distribution, such Fisher guarantees are much weaker than Kullback-Leibler guarantees. Bridging this gap requires some structural assumptions on the distribution.

Multiscale generative models. Images include structures at all scales, and several generative models have relied on decompositions with wavelet transforms (Yu et al., 2020; Gal et al., 2021). More recently, Marchand et al. (2022) established a connection between the renormalization group in physics and a conditional decomposition of the probability distribution of wavelet coefficients across scales. These models rely on maximum likelihood estimations with iterated Metropolis sampling, which leads to a high computational complexity.

Conditionally strongly log-concave distributions. We consider probability distributions whose Gibbs energy is dominated by quadratic interactions,

$$p(x) = \frac{1}{Z} \mathrm{e}^{-E(x)} \quad \text{with } E(x) = \frac{1}{2} x^{\mathrm{T}} K x + V(x).$$

The matrix K is positive symmetric and V is a non-quadratic potential. If V is non-convex, then p is a priori not log-concave. However, the Hessian of E may be dominated by the large

eigenvalues of K, whose corresponding eigenvectors define directions in which p is log-concave. For multiscale stationary distributions, K is a convolution whose eigenvalues have a power-law growth at high frequencies. As a result, the conditional distribution of high frequencies given lower frequencies may be log-concave.

Section 2.2 introduces factorizations of probability distributions into products of conditional distributions with arbitrary hierarchical projectors. If the projectors are adapted to obtain strongly log-concave factors, we prove that maximum likelihood estimation can be replaced by score matching, which is computationally more efficient. The MALA sampling algorithm also has a fast convergence due to the conditional log-concavity. Section 2.3 describes a class of multiscale physical processes that admit conditionally strongly log-concave (CSLC) decompositions with wavelet packet projections. This class includes the φ^4 model studied in statistical physics. These results thus provide an approach to provably avoid the numerical instabilities at phase transitions observed in such models. We then show in Section 2.4 that wavelet packet CSLC decompositions provide accurate models of cosmological weak lensing images, synthesized as test data for the Euclid outer-space telescope mission (Laureijs et al., 2011).

2.2 Conditionally strongly log-concave models

Section 2.2.1 introduces conditionally strongly log-concave models, by factorizing the probability density into conditional probabilities. For these models, Sections 2.2.2 and 2.2.3 give upper bounds on learning errors with score matching algorithms, and Section 2.2.4 on sampling errors with a Metropolis-Adjusted Langevin Algorithm (MALA). We omit the proofs of the mathematical results in this section, which were not done by the author of this manuscript. We refer the reader to the original publication (Guth et al., 2023a, Appendix E).

2.2.1 Conditional factorization and log-concavity

We introduce a probability factorization based on orthogonal projections on progressively smaller-dimensional spaces. The projections are adapted to define strongly log-concave conditional distributions.

Orthogonal factorization. Let $x \in \mathbb{R}^d$. A probability distribution $p(x)$ can be decomposed into a product of autoregressive conditional probabilities

$$p(x) = p(x[1]) \prod_{i=2}^{d} p(x[i] \,|\, x[1], \dots, x[i-1]). \tag{2.1}$$

However, more general factorizations can be obtained by considering blocks of variables in an orthogonal basis. We initialize the decomposition with $x_0 = x$. For $j = 1$ to J, we recursively split x_{j-1} in two orthogonal projections:

$$x_j = G_j x_{j-1} \text{ and } \bar{x}_j = \bar{G}_j x_{j-1},$$

where G_j and $\bar{G}_j$ are unitary operators such that $G_j^{\mathrm{T}} G_j + \bar{G}_j^{\mathrm{T}} \bar{G}_j = \mathrm{Id}$. It follows that

$$x_{j-1} = G_j^{\mathrm{T}} x_j + \bar{G}_j^{\mathrm{T}} \bar{x}_j. \tag{2.2}$$

Let $d_j = \dim(x_j)$ and $\bar{d}_j = \dim(\bar{x}_j)$, then $d_{j-1} = d_j + \bar{d}_j$.

Since the decomposition is orthogonal, for any probability distribution p we have

$$p(x_{j-1}) = p(x_j, \bar{x}_j) = p(x_j) p(\bar{x}_j | x_j).$$

Cascading this decomposition J times gives

$$p(x) = p(x_J) \prod_{j=1}^{J} p(\bar{x}_j | x_j), \tag{2.3}$$

which generalizes the autoregressive factorization (2.1). The properties of the factors $p(\bar{x}_j|x_j)$ depend on the choice of the orthogonal projectors G_j and $\bar{G}_j$, as we shall see below.

Model learning and sampling. A parametric model $p_\theta(x)$ of $p(x)$ can be defined from eq. (2.3) by computing parametric models of $p(x_J)$ and each $p(\bar{x}_j|x_j)$:

$$p_\theta(x) = p_{\theta_J}(x_J) \prod_{j=1}^{J} p_{\bar{\theta}_j}(\bar{x}_j | x_j), \tag{2.4}$$

with $\theta = (\theta_J, \bar{\theta}_j)_{j \geq J}$.

Learning this model then amounts to optimizing the parameters $\theta_J, (\bar{\theta}_j)_j$ from available data, so that the resulting distributions are close to the target. We measure the associated learning errors with the Kullback-Leibler divergences $\epsilon_J^L = \mathrm{KL}_{x_J}(p(x_J) \,\|\, p_{\theta_J}(x_J))$ and

$$\bar{\epsilon}_j^L = \mathbb{E}_{x_j}\Big[\mathrm{KL}_{\bar{x}_j}(p(\bar{x}_j|x_j) \,\|\, p_{\bar{\theta}_j}(\bar{x}_j|x_j))\Big],\ j \leq J.$$

Once the parameters have been estimated, we sample from p_θ as follows. We first compute a sample x_J of p_{θ_J}. The sampling introduces an error, which we measure with $\epsilon_J^S = \mathrm{KL}_{x_J}(\hat{p}_{\theta_J}(x_J) \,\|\, p_{\theta_J}(x_J))$, where $\hat{p}_{\theta_J}$ is the law of the samples returned by the algorithm. For each $j \leq J$, given the sampled x_j, we compute a sample $\bar{x}_j$ of $p_{\bar{\theta}_j}(\bar{x}_j|x_j)$ and recover x_{j-1} with eq. (2.2), up to $j = 1$, where it computes $x = x_0$. Let $\hat{p}_{\bar{\theta}_j}$ be the law of computed samples $\bar{x}_j$. It also introduces an error

$$\bar{\epsilon}_j^S = \mathbb{E}_{x_j}\Big[\mathrm{KL}_{\bar{x}_j}(\hat{p}_{\bar{\theta}_j}(\bar{x}_j|x_j) \,\|\, p_{\bar{\theta}_j}(\bar{x}_j|x_j))\Big],\ j \leq J.$$

Let $\hat{p}$ be the (joint) law of the computed samples x. The following proposition relates the total variation distance $\mathrm{TV}(\hat{p}, p)$ with the learning and sampling errors for each j.

Proposition 2.1 (Error decomposition)**.**

$$\mathrm{TV}(\hat{p}, p) \leq \frac{1}{\sqrt{2}} \left(\sqrt{\epsilon_J^L + \sum_{j=1}^{J} \bar{\epsilon}_j^L} + \sqrt{\epsilon_J^S + \sum_{j=1}^{J} \bar{\epsilon}_j^S} \right).$$

The overall error depends on the sum of learning and sampling errors for each conditional probability distribution. Therefore, to control the total error, we need sufficient conditions ensuring that each of these sources of error is small. We introduce CSLC models for this purpose.

Conditional strong log-concavity. We recall that a distribution p is strongly log-concave (SLC) if there exists $\beta[p] \geq \alpha[p] > 0$ such that

$$\alpha[p]\,\mathrm{Id} \preceq -\nabla_x^2 \log p(x) \preceq \beta[p]\,\mathrm{Id},\ \forall x. \tag{2.5}$$

Definition 2.1. *We say that* $p(x) = p(x_J) \prod_{j=1}^{J} p(\bar{x}_j|x_j)$ *is* conditionally strongly log-concave *(CSLC) if each* $p(\bar{x}_j|x_j)$ *is strongly log-concave in* $\bar{x}_j$ *for all* x_j*.*

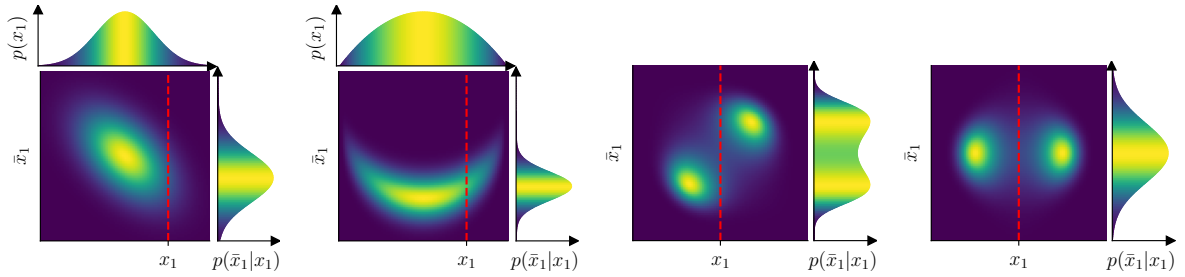

*FIGURE 2.1: A globally log-concave distribution is conditionally log-concave (*leftmost*), but the converse is not true (*middle left*): a non-convex support can have convex vertical slices (and horizontal projection). Conditional log-concavity also depends on the choice of orthogonal projectors: a distribution can fail to be conditionally log-concave in the canonical basis (*middle right*) but be conditionally log-concave after a rotation of* 45 *degrees (*rightmost*).*

Conditional log-concavity is a weaker condition than (joint) log-concavity. If $p(x)$ is log-concave, then it has a convex support. On the other hand, conditional log-concavity only constraints slices (through conditioning) and projections (through marginalization) of the support of $p(x)$. Figure 2.1 illustrates that a jointly log-concave distribution is conditionally log-concave (and $p(x_J)$ is furthermore log-concave), but the converse is not true. Conditional log-concavity also depends on the choice of the orthogonal projections G_j and $\bar{G}_j$ which need to be adapted to the data. A major issue is to identify projectors that define a CSLC decomposition, if it exists. We show in Section 2.3 that this can be achieved for a class of physical fields with wavelet packet projectors.

The following subsections provide bounds on the learning and sampling errors $\bar{\epsilon}_j^L$ and $\bar{\epsilon}_j^S$ for CSLC models. To simplify notations, in the following we drop the index j and replace $p_{\bar{\theta}_j}(\bar{x}_j|x_j)$ with $p_{\bar{\theta}}(\bar{x}|x)$. We shall suppose that the dimension $d_J = d(x_J)$ is sufficiently small so that x_J can be modeled and generated with any standard algorithm with small errors ϵ_J^L and ϵ_J^S ($d_J = 1$ in our numerical experiments).

2.2.2 Learning guarantees with score matching

Fitting probabilistic models $p_{\bar{\theta}}(\bar{x}|x)$ by directly minimizing the KL errors $\bar{\epsilon}^L$ is computationally challenging because of intractable normalization constants. Strong log-concavity enables efficient yet accurate learning via a tight relaxation to score matching.

There exist several frameworks to fit a parametric probabilistic model to the data, most notably the maximum-likelihood estimator of a general energy-based model $p_{\bar{\theta}}(\bar{x}|x) = Z_{\bar{\theta}}^{-1}(x)e^{-\bar{E}_{\bar{\theta}}(x,\bar{x})}$, where $\bar{E}_{\bar{\theta}}$ is an arbitrary parametric class. This is computationally expensive due to the need to estimate the gradients of the normalization constants $-\nabla_{\bar{\theta}} \log Z_{\bar{\theta}} = \mathbb{E}_{p_{\bar{\theta}}}[\nabla_{\bar{\theta}} \bar{E}_{\bar{\theta}}]$ during training, which requires the ability to sample from $p_{\bar{\theta}}(\bar{x}|x)$. An appealing alternative which has enjoyed recent popularity is *score matching* (Hyvärinen and Dayan, 2005), which instead minimizes the Fisher Divergence FI:[1]

$$\begin{aligned}\ell(\bar{\theta}) &= \mathbb{E}_x[\mathrm{FI}_{\bar{x}}(p(\bar{x}|x) \,\|\, p_{\bar{\theta}}(\bar{x}|x))] \\ &= \mathbb{E}_{x,\bar{x}}\left[\frac{1}{2}\|-\nabla_{\bar{x}} \log p(\bar{x}|x) - \nabla_{\bar{x}} \bar{E}_{\bar{\theta}}(x,\bar{x})\|^2\right].\end{aligned}$$

With a change of variables we obtain

$$\ell(\bar{\theta}) = \mathbb{E}_{x,\bar{x}}\left[\frac{1}{2}\|\nabla_{\bar{x}} \bar{E}_{\bar{\theta}}\|^2 - \Delta_{\bar{x}} \bar{E}_{\bar{\theta}}\right] + \mathrm{cst}, \tag{2.6}$$

[1] Our notation differs from the conventional use by a factor 1/2 for convenience.

showing that $\ell(\bar{\theta})$ can be minimized from available samples without estimating normalizing constants or sampling from $p_{\bar{\theta}}$. Indeed, given i.i.d. samples $\{(\bar{x}^1, x^1), \dots, (\bar{x}^n, x^n)\}$ from $p(\bar{x}, x)$, the empirical risk $\hat{\ell}(\bar{\theta})$ associated with score matching on $p(\bar{x}|x)$ is given by

$$\hat{\ell}(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{1}{2}\|\nabla_{\bar{x}}\bar{E}_{\bar{\theta}}(x^i, \bar{x}^i)\|^2 - \Delta_{\bar{x}}\bar{E}_{\bar{\theta}}(x^i, \bar{x}^i)\Big). \tag{2.7}$$

The score-matching objective avoids the computational barriers associated with normalization and sampling in high-dimensions, at the expense of defining a weaker metric than the KL divergence. This weakening of the metric is quantified by the log-Sobolev constant $\rho[p]$ associated with p. It is the largest $\rho > 0$ such that $\mathrm{KL}(q \,\|\, p) \leq \frac{1}{\rho}\mathrm{FI}(q \,\|\, p)$ for any q. Learning via score matching can therefore be seen as a relaxation of maximum-likelihood training, whose tightness is controlled by the log-Sobolev constant of the hypothesis class (Koehler et al., 2022). This constant can be exponentially small for general multimodal distributions, making this relaxation too weak. A crucial exception, however, is given by SLC distributions (or small perturbations of them), as shown by the Bakry-Emery criterion (Bakry et al., 2014, Definition 1.16.1): if $\alpha[p_{\bar{\theta}}(\bar{x}|x)] \geq \bar{\alpha} > 0$ for all x, or equivalently if $\nabla^2_{\bar{x}}\bar{E}_{\bar{\theta}} \succeq \bar{\alpha}\,\mathrm{Id}$ for all $x, \bar{x}$, then $\rho[p_{\bar{\theta}}(\bar{x}|x)] \geq \bar{\alpha}$ for all x, and therefore

$$\bar{\epsilon}^L \leq \frac{1}{\bar{\alpha}}\ell(\bar{\theta}). \tag{2.8}$$

We remark that while eq. (2.8) does not make explicit CSLC assumptions on the reference distribution p, a consistent learning model implies that the conditional distribution $p(\bar{x}|x)$ is arbitrarily well approximated (in KL divergence) with SLC distributions—thus justifying the structural CSLC assumption on the target.

2.2.3 Score matching with exponential families

In numerical applications, one cannot minimize the true score-matching loss ℓ as only a finite amount of data is available. We now show that a similar control as eq. (2.8) can be obtained for the empirical loss minimizer, whenever prior information enables us to define low-dimensional exponential models for $p_{\bar{\theta}}(\bar{x}|x)$ with good accuracy. It also provides a control on the critical parameter $\bar{\alpha}$, addressing the optimization and statistical errors.

We consider a linear model $\bar{E}_{\bar{\theta}}(x, \bar{x}) = \vec{\theta}^{\mathrm{T}}\bar{\Phi}(x, \bar{x})$ with a fixed potential vector $\bar{\Phi}(x, \bar{x}) \in \mathbb{R}^m$ (m is thus the number of parameters), and the corresponding minimization of the (conditional) score matching objective in eq. (2.7). Thanks to this linear parameterization, it becomes a convex quadratic form $\hat{\ell}(\bar{\theta}) = \frac{1}{2}\vec{\theta}^{\mathrm{T}}\hat{H}\vec{\theta} - \vec{\theta}^{\mathrm{T}}\hat{g}$, with

$$\hat{H} = \frac{1}{n}\sum_{i=1}^{n}\nabla_{\bar{x}}\bar{\Phi}(x^i, \bar{x}^i)\nabla_{\bar{x}}\bar{\Phi}(x^i, \bar{x}^i)^{\mathrm{T}} \in \mathbb{R}^{m\times m},$$
$$\hat{g} = \frac{1}{n}\sum_{i=1}^{n}\Delta_{\bar{x}}\bar{\Phi}(x^i, \bar{x}^i) \in \mathbb{R}^m.$$

It can be minimized in closed-form by inverting the Hessian matrix: $\hat{\bar{\theta}} = \hat{H}^{-1}\hat{g}$. As discussed, the sampling and learning guarantees of the model critically rely on the CSLC property, which is ensured as long as $\hat{\bar{\theta}} \in \Theta_{\bar{\alpha}} := \{\bar{\theta} \,|\, \nabla^2_{\bar{x}}\bar{E}_{\bar{\theta}}(x, \bar{x}) \succeq \bar{\alpha}\,\mathrm{Id},\ \forall\,(x, \bar{x})\}$ with $\bar{\alpha} > 0$.

The following theorem leverages the finite-dimensional linear structure of the score-matching problem to establish fast non-asymptotic rates of convergence, controlling the excess risk *in* KL *divergence*.

Theorem 2.1 (Excess risk for CSLC exponential models)**.** *Let* $\bar{\theta}^\star = \arg\min \ell(\bar{\theta})$ *and* $\hat{\bar{\theta}} = \arg\min \hat{\ell}(\bar{\theta})$*. Assume:*

(i) $\bar{\theta}^\star \in \Theta_{\bar{\alpha}}$ *for some* $\bar{\alpha} > 0$,

(ii) $H = \mathbb{E}\Big[\nabla_{\bar{x}}\bar{\Phi}\nabla_{\bar{x}}\bar{\Phi}^{\mathrm{T}}\Big] \succeq \eta\,\mathrm{Id}$ *with* $\eta > 0$,

(iii) the sufficient statistics $\bar{\Phi}$ *satisfy* $\nabla\bar{\Phi}_k(x,\bar{x})$ *is* $M_{\bar{\Phi}}$*-Lipschitz for any* $k \leq m$ *and all* x*, as well as moment and regularity conditions detailed in Guth et al. (2023a, Appendix E).*

Then when $n > m$*, the empirical risk minimizer* $\hat{\bar{\theta}}$ *satisfies*

$$\hat{\bar{\theta}} \in \Theta_{\hat{\bar{\alpha}}} \text{ with } \mathbb{E}_{(\bar{x}^i,x^i)}\Big[\hat{\bar{\alpha}}\Big] \geq \bar{\alpha} - O\Big(\eta^{-1}\sqrt{\frac{m}{n}}\Big), \tag{2.9}$$

and, for $t \ll \sqrt{m}\ell(\bar{\theta}^\star)$*,*

$$\bar{\epsilon}^L \leq \frac{\ell(\bar{\theta}^\star)}{\bar{\alpha}}(1+t) \tag{2.10}$$

with probability greater than $1 - \exp\{-O(n\log(tn/\sqrt{m}))\}$ *over the draw of the training data. The constants in* $O(\cdot)$ *only depend on moment and regularity properties of* $\bar{\Phi}$*.*

The theorem provides learning guarantees for the empirical risk minimizer $\hat{\bar{\theta}}$ (compare Equations (2.8) and (2.10)), and hinges on three key properties: the ability of the exponential family to approximate the true conditionals at each block (i) with small Fisher approximation error $\ell(\bar{\theta}^\star)$, (ii) with a sufficiently large strong log-concavity parameter $\bar{\alpha}$, and (iii) with a well-conditioned kernel H. In numerical applications, the number of parameters m should be small enough to control the learning error for finite number of samples n, and to be able to compute and invert the Hessian matrix $\hat{H}$. We will define in Section 2.3 low-dimensional models that can approximate a wide range of multiscale physical fields.

The proof uses concentration of the empirical covariance $\hat{H}$, and combines both upper and lower tail probability bounds (Mourtada, 2022; Vershynin, 2012) to bound the expectation, similarly as known results for least-squares (Mourtada, 2022; Hsu et al., 2012). The statistical properties of score matching under exponential families have been studied in the infinite-dimensional setting by Sriperumbudur et al. (2013); Sutherland et al. (2018), where kernel ridge estimators achieve non-parametric rates n^{-s}, $s < 1$. Compared to these, as an intermediate result, we achieve the optimal rate in FI divergence in n^{-1} directly with the ridgeless estimator. The key assumption is *(i)*, namely that the optimal model in the exponential family is SLC. Since our structural assumption on the target p is precisely that its conditionals are SLC, it is reasonable to expect this to be generally true. For instance, this is the case if the model is well specified ($p = p_{\bar{\theta}^\star}$).

2.2.4 Sampling guarantees with MALA

We illustrate the efficient sampling properties of CSLC distributions by focusing on a reference sampler given by the Metropolis-Adjusted Langevin Algorithm (MALA) with algorithmic warm-start, which enjoys well-understood convergence properties in this case:

Proposition 2.2 (MALA Sampling, Altschuler and Chewi (2023, Theorem 5.1))**.** *Suppose that* $\bar{\alpha}\,\mathrm{Id} \preceq \nabla^2_{\bar{x}}\bar{E}_{\bar{\theta}}(\bar{x}|x) \preceq \bar{\beta}\,\mathrm{Id}$ *for all* $\bar{x}, x$*, and let* $\bar{d} = \dim(\bar{x})$*. Then* N *steps of MALA produce a sample* $\bar{x}$ *with conditional law* $\hat{p}_{\bar{\theta}}(\bar{x}|x)$ *satisfying*

$$\bar{\epsilon}^S \leq \exp\left(-O\left(\sqrt{\frac{N}{\sqrt{\bar{d}}\bar{\beta}/\bar{\alpha}}}\right)\right).$$

MALA can thus be used to sample from CSLC distributions with an exponential convergence, whose mixing time $\widetilde{O}(\sqrt{\bar{d}}\bar{\beta}/\bar{\alpha})$ is sublinear in the dimension $\bar{d}$ and linear in the condition

number $\bar{\beta}/\bar{\alpha}$ of the Hessian $\nabla^2_{\tilde{x}}\bar{E}_{\bar{\theta}}$. We also note that similar guarantees will hold for other high-precision Metropolis-Hastings samplers, such as Hamilton Monte-Carlo. Together, Propositions 2.1 and 2.2 and Theorem 2.1 imply a control on the total accumulated error for CSLC exponential models.

2.3 Wavelet packet conditional log-concavity

The CSLC property depends on the choice of the projectors $(\bar{G}_j, G_j)$ which need to be adapted to the data. We show that for a class of stationary multiscale physical processes, CSLC models can be obtained with wavelet packet projectors. These models exploit the dominating quadratic interactions at high frequencies by splitting the frequency domain in sufficiently narrow bands. It reveals a powerful mathematical structure in this class of complex distributions.

2.3.1 Energies with scalar potentials

In the following, $x \in \mathbb{R}^d$ is a $\sqrt{d} \times \sqrt{d}$ image or two-dimensional field. We denote $x[i]$ the value of x at pixel or location i. An important class of stationary probability distributions $p(x) = Z^{-1}\mathrm{e}^{-E(x)}$ are defined in physics from an energy composed of a two-point interaction term K plus a potential that is a sum of scalar potentials v:

$$E(x) = \frac{1}{2}x^{\mathrm{T}}Kx + \sum_i v(x[i]). \tag{2.11}$$

The matrix K is a positive symmetric convolution operator. Equation (2.11) generalizes both zero-mean Gaussian processes (if $v = 0$ then K is the inverse covariance) and distributions with i.i.d. components (if $K = 0$ then v is the negative log-density of the pixel values). The energy Hessian is given by

$$\nabla^2_x E(x) = K + \mathrm{diag}\Big(v''(x[i])\Big)_i. \tag{2.12}$$

If $v''(t) < 0$ for some $t \in \mathbb{R}$ then we may get negative eigenvalues for some x, in which case the energy is not convex.

Equation (2.11) provides models of a wide class of physical phenomena (Marchand et al., 2022), including ferromagnetism. An important example is the φ^4 energy in physics, which is a non-convex energy allowing to study phase transitions and explain the nature of numerical instabilities (Zinn-Justin, 2021). It has a kinetic energy term defined by $K = -\beta\Delta$ where Δ is a discrete Laplacian that enforces spatial regularity, and its scalar potential is $v(t) = t^4 - (1+2\beta)t^2$. It has a double-well shape which pushes the values of each $x[i]$ towards $+1$ and -1, and is thus non-convex. β is an inverse temperature parameter. In the thermodynamic limit $d \to \infty$ of infinite system size, the φ^4 energy has a phase transition at $\beta_c \approx 0.68$ (Kaupužs et al., 2016). At small temperature ($\beta \geq \beta_c$), the local interactions in the energy give rise to long-range dependencies. Gibbs sampling then "critically slows down" (Chaikin et al., 1995; Sethna, 2021) due to these long-range dependencies.

Fast sampling can nevertheless be obtained by exploiting conditional strong log-concavity. Assume that there exists $\gamma > 0$ such that $v''(t) \geq -\gamma$ for all $t \in \mathbb{R}$. It then follows that $\nabla^2_x E \succeq K - \gamma\,\mathrm{Id}$. We can thus obtain a convex energy by restricting K over a subspace where its eigenvalues are larger than γ. The convolution K is diagonalized by the Fourier transform, with positive eigenvalues that we write $\hat{K}(\omega)$ at all frequencies ω. The value $\hat{K}(\omega)$ typically increases when the frequency modulus $|\omega|$ increases. A convex energy is then obtained with a projector over a space of high-frequency images, as shown in the following proposition.

Proposition 2.3 (Conditional log-concavity of scalar potential energies)**.** *Consider the energy defined in eq. (2.11) and assume that $-\gamma \leq v'' \leq \delta$ for some $\gamma, \delta > 0$ and that $\hat{K}(\omega) = \lambda|\omega|^\eta$ for*

some $\eta > 0$. Let $\bar{G}_1$ be an orthogonal projector over a space of signals whose Fourier transform have a support included over frequencies ω such that $|\omega| \geq |\omega_0|$ with $|\omega_0| > (\gamma/\lambda)^{1/\eta}$. Then the conditional probability $p(\bar{x}_1|x_1)$ is strongly log-concave for all x_1.

The proof is in Appendix A.5 and relies on a direct calculation of the Hessian of the conditional energy. This proposition proves that we obtain a strongly log-concave conditional distribution $p(\bar{x}_1|x_1)$ with a sufficiently high-frequency filter $\bar{G}_1$. It is illustrated in the two rightmost panels of Figure 2.1 on a simplified two-dimensional example inspired from the φ^4 energy. The distribution has two modes $x = (1, 1)$ and $x = (-1, -1)$, and the Fourier coefficients are computed with a 45 degrees rotation: $x_1 = (x[1] + x[2])/\sqrt{2}$ and $\bar{x}_1 = (x[2] - x[1])/\sqrt{2}$, which leads to a log-concave conditional distribution.

Multiscale physical fields with scalar potential energies (2.11) are often self-similar over scales, in the sense that lower-frequency fields x_j can also be described with an energy in the form of eq. (2.11), with different parameters (Wilson, 1971). This explains why Proposition 2.3 can be iterated to obtain a CSLC decomposition. For φ^4 energies, the range of $\bar{G}_1$ is non-empty as soon as $\beta \geq \frac{1}{2}$, which includes the critical temperature $\beta_c \approx 0.68$ (though $\delta = \infty$). At the critical temperature, x_1 is further described by the same parameters K and v as x, so that a complete CSLC decomposition is obtained by iteratively selecting projectors $\bar{G}_j$ which isolate the highest frequencies of x_{j-1}.

Proposition 2.3 can be extended to general energies

$$E(x) = \frac{1}{2} x^{\mathrm{T}} K x + V(x),$$

by assuming that the Hessian $\nabla^2 V(x)$ is bounded above and below. Conditional log-concavity may then be found by exploiting dominating quadratic energy terms with a PCA of K. We believe that this general principle may hold beyond the case of scalar potential energies (2.11) considered here.

2.3.2 Wavelet packets and renormalization group

We now define wavelet packet projectors G_j and $\bar{G}_j$, which are orthogonal projectors on localized zones of the Fourier plane. They are computed by convolutions with conjugate mirror filters and subsamplings Coifman et al. (1992), described in Appendix A.1. These filters perform a recursive split of the frequency plane illustrated in Figure 2.2.

The wavelet packet $\bar{G}_j$ is a projector on a high-frequency domain, whereas G_j is a projection on the remaining lower-frequency domain. An orthogonal wavelet transform is a particular example, which decomposes the Fourier plane into annuli of about one octave bandwidth, as shown in the top left and bottom panels of Figure 2.2. However, it may not be sufficiently well localized in the Fourier domain to obtain strictly convex energies. The frequency localization is improved by refining this split, as illustrated on the top right panel of Figure 2.2. Each $\bar{G}_j$ then performs a projection over a frequency annulus whose bandwidth is a half octave. Wavelet packets can adjust the frequency bandwidth to 2^{-M+1} octave for any integer $M \geq 1$. It allows reducing the support of $\bar{G}_j$, which is necessary to obtain a CSLC decomposition according to Proposition 2.3.

2.3.3 Multiscale scalar potentials

The probability distribution $p(x)$ is approximated by $p_\theta(x) = p_{\theta_J}(x_J) \prod_{j=1}^{J} p_{\bar{\theta}_j}(\bar{x}_j|x_j)$, where each x_j and $\bar{x}_j$ are computed with wavelet packet projectors G_j and $\bar{G}_j$. We introduce a parameterization of $p_{\bar{\theta}_j}$ with scalar potential energies, following Marchand et al. (2022). We shall suppose that the dimension $d_J = \dim(x_J)$ is sufficiently small so that $p(x_J)$ may be approximated with any standard algorithm ($d_J = 1$ in our numerical experiments).

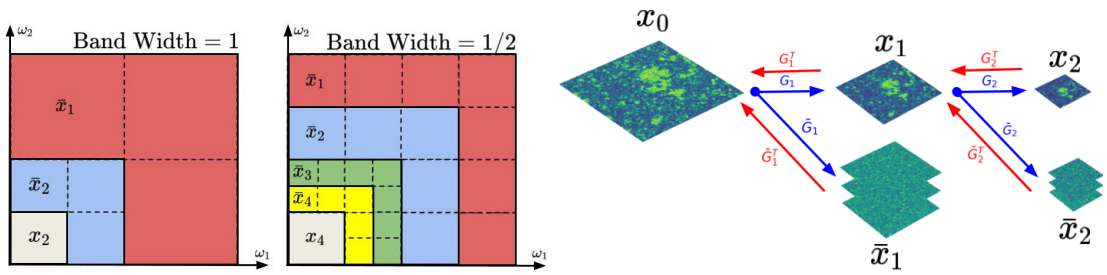

FIGURE 2.2: Left: *frequency localization of the decomposition* $(x_J, \bar{x}_J, \ldots, \bar{x}_1)$ *with wavelet packet projectors of* 1 *(*left*) and* 1/2 *(*right*) octave bandwidths.* Right: *iterative decomposition of* $x = x_0$ *with* $(\tilde{G}_j, G_j)$ *implementing a wavelet packet transformation over* $J = 2$ *layers of* 1 *octave bandwidth.*

The self-similarity property of multiscale fields with scalar energies motivates the definition of each $p_{\bar{\theta}_j}(\bar{x}_j|x_j)$ with an interaction energy

$$\begin{aligned}\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j) &= \frac{1}{2}\bar{x}_j^{\mathrm{T}}\bar{K}_j\bar{x}_j + \bar{x}_j^{\mathrm{T}}\bar{K}'_j x_j + \sum_i \bar{v}_j(x_{j-1}[i]) \\ &= \bar{\theta}_j^T \bar{\Phi}_j(x_j, \bar{x}_j),\end{aligned} \tag{2.13}$$

which derives from the fact that $p(x_{j-1})$ defines an energy of the form (2.11) (Marchand et al., 2022). $\bar{\Phi}_j$ captures the interaction terms and performs a parametrized approximation of $\bar{v}_j$, defined in Appendix A.2.1.

The parameters $\bar{\theta}_j$ are estimated from samples by inverting the empirical score matching Hessian as in Section 2.2.3. We generate samples from the resulting distribution p_θ by sampling from p_{θ_J} and then iteratively from each $p_{\bar{\theta}_j}$ with MALA. The learning and sampling algorithms are summarized in Appendix A.2.2. Additionally, Appendix A.4 explains that a parameterized model of the global energy (2.11), which is crucial for scientific applications, can be recovered with free-energy score matching.

2.4 Numerical results

This section demonstrates that a wavelet packet decomposition of φ^4 scalar fields and weak-lensing cosmological fields defines strongly log-concave conditional distributions. It allows efficient learning and sampling algorithms, and leads to higher-resolution generations than in previous works.

2.4.1 φ^4 scalar potential energy

We learn a wavelet packet model of φ^4 scalar fields at different temperatures, using the decomposition and models presented in Section 2.3. The wavelet packet exploits the conditionally strongly log-concave property of φ^4 scalar fields (Proposition 2.3) to obtain a small error in the generated samples, as shown in Section 2.2. We first verify qualitatively and quantitatively that this error is small.

We evaluate the wavelet packet model at three different temperatures, which have different statistical properties: $\beta = 0.50$, the "disorganized" state, $\beta = 0.68 \approx \beta_c$ the critical point, and $\beta = 0.76$ the "organized" state. The computational efficiency of our approach enables generating high-resolution 128×128 images, as opposed to 32×32 in Marchand et al. (2022). Indeed, learning the model parameters for 64×64 images with score matching takes seconds on GPU, whereas doing the same with maximum likelihood takes hours on CPU (as sequential MCMC steps are not easily parallelized). The generated samples are shown in Figure 2.3 and

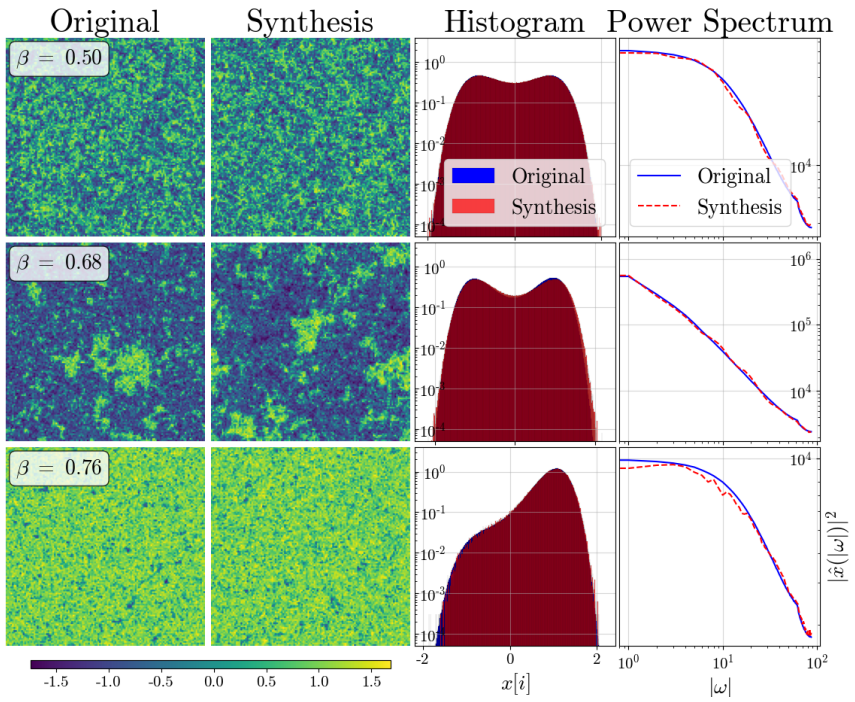

FIGURE 2.3: Comparison between training and generated samples for φ^4 energies. In columns: *training samples, generated samples, histograms of marginal distributions $p(x[i])$ and power spectrum.* In rows: *disorganized state $\beta = 0.50$, critical point $\beta = 0.68 \approx \beta_c$, and organized state $\beta = 0.76$.*

are qualitatively indistinguishable from the training data. The experimental setting is detailed in Appendix A.3.

A distribution $p(x)$ having a scalar potential energy (2.11) is a maximum-entropy distribution constrained by second-order moments and hence by the power spectrum, and by the marginal distribution of all $x[i]$. These statistics specify the matrix K and the scalar potential $v(t)$. Our model p_θ also has a scalar potential energy in this case. To guarantee that $p_\theta = p$, it is thus sufficient to show that they have the same power spectrum and same marginal distributions. We perform a quantitative validation of generated samples by comparing their marginal densities and Fourier spectrum with the training data. Figure 2.3 shows that these statistics are well recovered by our model.

2.4.2 Conditional log-concavity

We numerically verify that φ^4 at critical temperature is CSLC (Definition 2.1), with appropriate wavelet packet projectors. It amounts to verifying that the eigenvalues of the conditional Hessian $\nabla^2_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j)$ are positive for all x_j and $\bar{x}_j$. We can restrict x_j to typical samples from $p(x_j)$. However, it is important that the Hessian be positive even for $\bar{x}_j$ outside of the support of $p(\bar{x}_j|x_j)$. Indeed, negative eigenvalues occur at local directional maxima of the energy, rather than minima which would correspond to most likely samples. We thus evaluate the Hessian at $\bar{x}_j = 0$, which is expected to be such an adversarial point.

Figure 2.4 shows distributions of eigenvalues of $\nabla^2_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}$ for decompositions $(\bar{G}_j, G_j)$ of various frequency bandwidths. It shows that the smallest eigenvalues become larger and eventually cross zero as the frequency bandwidth of $\bar{G}_j$ becomes narrower, as predicted by Proposition 2.3. Furthermore, the condition number of the Hessian becomes smaller as eigenvalues concentrate towards their mean.

As shown in eq. (2.12), both the quadratic part K and the scalar potential v contribute to the Hessian. As a way to visualize both contributions, we define the equivalent scalar potential v^0

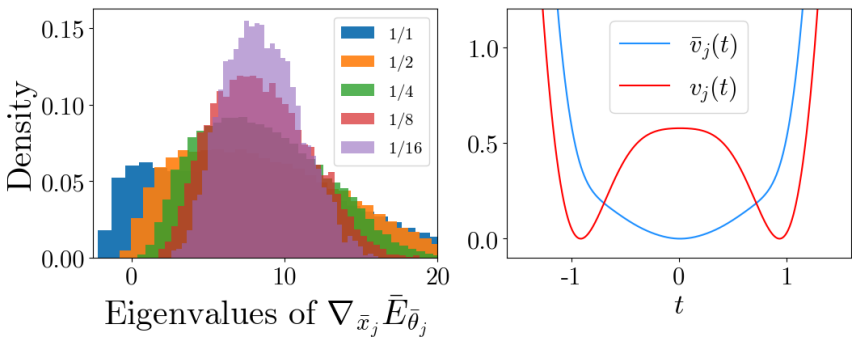

FIGURE 2.4: Conditional strong log-concavity of φ^4 at critical temperature. All scales j yield similar results. Left: *distribution of eigenvalues of $\nabla^2_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}$ for different frequency bandwidths ($j = 1$ is shown).* Right: *equivalent scalar potentials v_j and $\bar{v}_j$ ($j = 3$ is shown).*

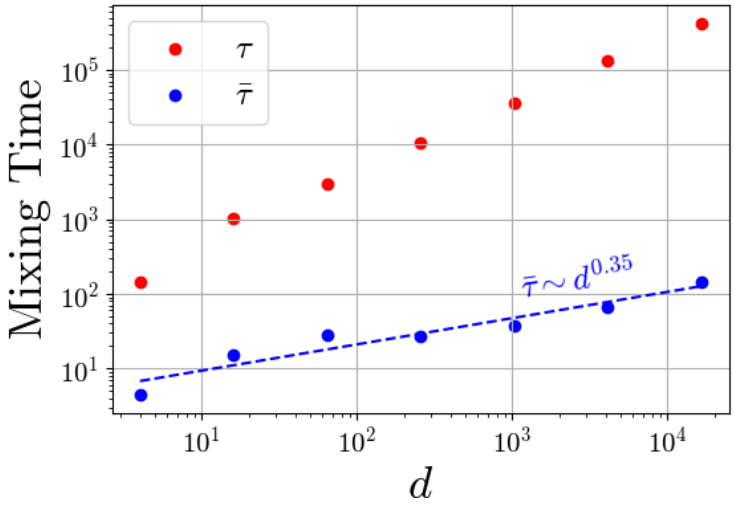

FIGURE 2.5: Mixing times for direct (τ) and conditional ($\bar{\tau}$) sampling for φ^4 at critical temperature.

as $v^0(t) = v(t) + \frac{\mathrm{Tr}(K)}{2d} t^2$. It corresponds to extracting the mean quadratic value $\mathrm{Tr}(K)/2d \, \|x\|^2$ from the quadratic part and reinterpreting it as a scalar potential. This allows visualizing the average energy on a pixel value when neglecting spatial correlations. The right panel of Figure 2.4 compares these equivalent scalar potentials for the energy E_j of x_j and the conditional energy $\bar{E}_j$. It shows that the non-convex double-well potential in the global energy becomes convex after the conditioning. It verifies Proposition 2.3, as the mean quadratic value becomes larger when we restrict K to a subspace of high-frequency signals.

We also verify the sampling efficiency predicted by Proposition 2.2. As we cannot evaluate the KL divergences $\bar{\epsilon}^S_j$, we rather compute the decorrelation mixing time $\bar{\tau}$, a measure of the number of steps of conditional MALA to reach a given fixed error threshold averaged over all scales j. The precise definition is given in Appendix A.3.3. We compare it with the decorrelation mixing time τ of MALA on the non-convex global energy E.

Sampling maps of size $\sqrt{d} \times \sqrt{d}$ from the global φ^4 energy E at the critical temperature requires a number of steps $\tau \sim d^{1.0}$ (Zinn-Justin, 2021). This phenomena is known as critical slowing down (Chaikin et al., 1995; Sethna, 2021), a consequence of long-range correlations. We numerically show that our algorithm does not suffer from it. Figure 2.5 indeed demonstrates an empirical scaling $\bar{\tau} \sim d^{0.35}$. Note that this is not directly comparable with Proposition 2.2 as the decorrelation mixing time defines a different convergence rate than the KL mixing time.

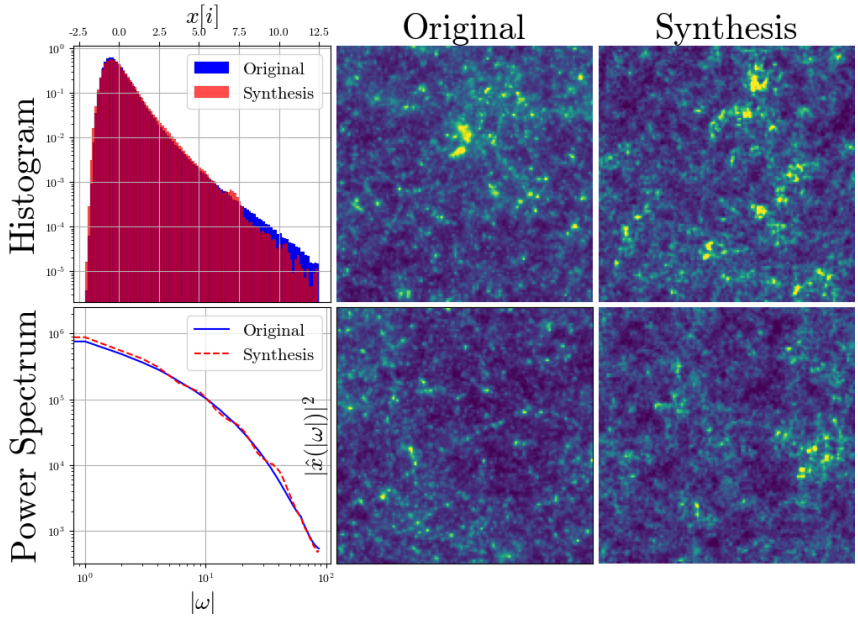

FIGURE 2.6: Comparison between training and generated samples for weak-lensing maps. Upper left: *histograms of marginal distributions* $p(x[i])$. Lower left: *power spectrum.* Center: *training samples.* Right: *generated samples.*

2.4.3 Application to cosmological data

We now apply our algorithm to generate high-resolution weak lensing convergence maps (Bartelmann and Schneider, 2001; Kilbinger, 2015) with an explicit probability model. Weak lensing convergence maps measure the bending of light near large gravitational masses on two-dimensional slices of the universe. We used simulated convergence maps computed by the Columbia lensing group (Zorrilla Matilla et al., 2016; Gupta et al., 2018) as training data. They simulate the next generation outer-space telescope *Euclid* of the European Space Agency (Laureijs et al., 2011), which will be launched in 2023 to accurately determine the large scale geometry of the universe governed by dark matter. Estimating the probability distribution of such maps is therefore an outstanding problem (Marchand et al., 2022). We demonstrate that the CSLC property is surprisingly verified in this real-world example, and can be used to efficiently model and generate these complex fields.

We use the same models and algorithms as for the φ^4 energy. The experimental setting is detailed in Appendix A.3. Figure 2.6 shows that our generated samples are visually highly similar to the training data. Quantitatively, they have nearly the same power spectrum. The marginal distribution of all $x[i]$ are also nearly the same, with a long tail corresponding to high amplitude peaks, which are typically difficult to reproduce. As opposed to microcanonical simulations with moment-matching algorithms (Cheng and Ménard, 2021), we compute an explicit probability distribution model, which is exponential. As a maximum-entropy model, it has a higher entropy than the true distribution, and therefore does not suffer from lack of diversity. By relying on the CSLC property, we can use the fast score-matching algorithm and compute 128×128 images, at four times the 32×32 resolution than with a maximum-likelihood algorithm used in Marchand et al. (2022).

Figure 2.7 shows the equivalent scalar potentials of the conditional energies at all scales, which are all convex and thus verify the CSLC property of weak lensing model. It demonstrates that this property can be used to efficiently model and generate high-resolution complex data.

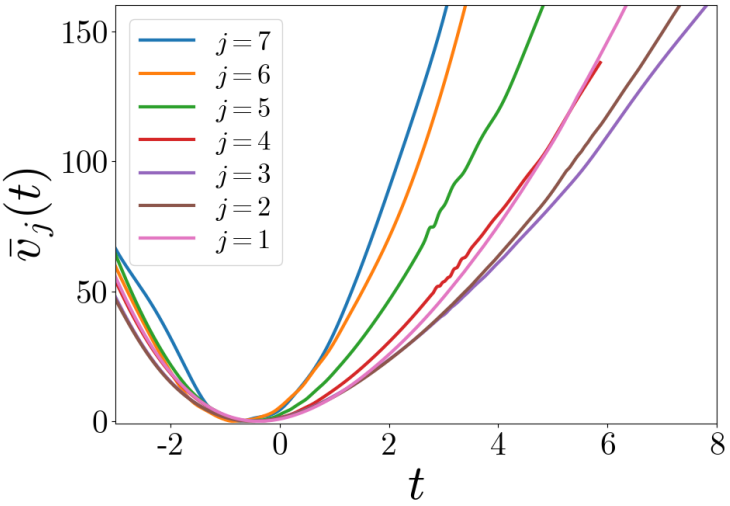

FIGURE 2.7: Equivalent scalar potentials $\bar{v}_j$ at each scale j for weak-lensing maps (normalized for viewing purposes).

2.5 Discussion

We introduced conditionally strongly log-concave (CSLC) models and proved that they lead to efficient learning with score matching and sampling with MALA, while controlling errors. These models rely on iterated orthogonal projections of the data that are adapted to its distribution. We showed mathematically and numerically that complex multiscale physical fields satisfy the CSLC property with wavelet packet projectors. The argument is general and relies on the presence of a quadratic (kinetic) energy term which ensures strong log-concavity at high-frequencies. It provides high-quality and efficient generation of high-resolution fields even when the underlying distribution is unknown. The CSLC property guarantees diverse generations without memorization issues, which is critical in scientific applications.

CSLC models can be extended by introducing latent variables. The guarantees of Section 2.2 extend to the case where the data is a marginal of a CSLC distribution. A notable example is a score-based diffusion model, for which the data $x = x_0$ is a marginal of a higher-dimensional process $(x_t)_t$ whose conditionals $p(x_{t-\delta}|x_t)$ are approximately Gaussian white when δ is small, thus introducing a tradeoff between the number of terms in the CSLC decomposition and the condition number of its factors. Score diffusion is a generic transformation, but it assumes that the score $\nabla_{x_t} \log p(x_t)$ can be estimated with deep networks at any $t \geq 0$ (Song et al., 2021b; Ho et al., 2020). For high-resolution images, the score estimation often uses conditional multiscale decompositions (Saharia et al., 2021; Ho et al., 2022; Dhariwal and Nichol, 2021). Understanding the log-concavity properties of natural image distributions under such transformations is a promising research avenue to understand the effectiveness of score-based diffusion models. We now turn to score-based diffusion models in Chapters 3 and 4.

Chapter

3 Wavelet Score-Based Generative Models

Chapter content

3.1 Introduction 46
3.2 Sampling and discretization of score-based generative models . . . 47
3.2.1 Score-based generative models 47
3.2.2 Discretization of SGMs and score regularity 48
3.3 Wavelet score-based generative models 50
3.3.1 Wavelet whitening and cascaded SGMs 50
3.3.2 Discretization and accuracy for Gaussian processes 52
3.4 Acceleration with WSGM: numerical results 53
3.4.1 Physical processes with scalar potentials 53
3.4.2 Scale-wise time reduction in natural images 54
3.5 Discussion 56

We have shown in Chapter 2 that multiscale physical fields have log-concave wavelet conditional distributions. This may not be true for more complex distributions such as natural images or faces. Despite this, score-based generative models manage to generate high-quality samples from these distributions.

A drawback of these models is that the discretization of the reverse SDE typically requires a large number of time steps and hence a high computational cost. This has been partially alleviated by multiscale generation approaches, which are implicitly performing a conditional factorization of the data probability distribution that is similar to the one introduced by Marchand et al. (2022) and studied in Chapter 2. We explain how this acceleration results from better regularity properties of the conditional wavelet scores as opposed to the (joint) global score. The resulting Wavelet Score-based Generative Model (WSGM) synthesizes wavelet coefficients with the same number of time steps at all scales, and its time complexity therefore grows linearly with the image size. This is proved mathematically for Gaussian distributions, and shown numerically for the φ^4 model and a celebrity face dataset.

This chapter is adapted from the following publication: Florentin Guth, Simon Coste, Valentin De Bortoli, and Stéphane Mallat. Wavelet score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2022. We omit the proofs of the mathematical results, which were not done by the author of this dissertation. This chapter was written before Chapters 2 and 4, and is therefore less mature. In particular, the theoretical analysis of the discretization of score-based diffusions was not as developed, as this chapter predates Chen et al. (2022b,b). It was then apparent that the central theoretical issue was rather to study the estimation of the scores. This problem was dealt with in Chapter 2 by relying on conditional log-concavity to avoid the need for score-based diffusion models, and leveraging prior information to obtain low-dimensional parametric models of the conditional energies. We present preliminary

results on the topic of score estimation for score-based diffusion models of natural images in Chapter 4.

3.1 Introduction

Score-based Generative Models (SGMs) have obtained remarkable results to learn and sample probability distributions of image and audio signals (Song and Ermon, 2019; Chen et al., 2021; Kong et al., 2021; Nichol and Dhariwal, 2021; Popov et al., 2021; Dhariwal and Nichol, 2021). They proceed as follows: the data distribution is mapped to a Gaussian white distribution by evolving along a Stochastic Differential Equation (SDE), which progressively adds noise to the data. The generation is implemented using the time-reversed SDE, which transforms a Gaussian white noise into a data sample. At each time step, it pushes samples along the gradient of the log probability, also called *score function*. This score is estimated by leveraging tools from score-matching and deep neural networks (Hyvärinen and Dayan, 2005; Vincent, 2011). At sampling time, the computational complexity is therefore proportional to the number of time steps, i.e., the number of forward network evaluations. Early SGMs in Song and Ermon (2019); Song et al. (2021b); Ho et al. (2020) used thousands of time steps, and hence had a limited applicability.

Diffusion models map a Gaussian white distribution into a highly complex data distribution. We thus expect that this process will require a large number of time steps. It then comes as a surprise that recent approaches have drastically reduced this time complexity. This is achieved by optimizing the discretization schedule or by modifying the original SGM formulation (Kadkhodaie and Simoncelli, 2021; Jolicoeur-Martineau et al., 2021; Liu et al., 2022a; Zhang and Chen, 2022; San-Roman et al., 2021; Nachmani et al., 2021; Song et al., 2020; Kong and Ping, 2021; Ho et al., 2020; Luhman and Luhman, 2021; Salimans and Ho, 2022; Xiao et al., 2021). High-quality score-based generative models have also been improved by cascading multiscale image generations (Saharia et al., 2021; Ho et al., 2022; Dhariwal and Nichol, 2021) or with subspace decompositions (Jing et al., 2022). We make explicit the reason of this improvement, which provably accelerates the sampling of SGMs.

A key idea is that typical high-dimensional probability distributions coming from physics or natural images have complex multiscale properties. They can be simplified by factorizing them as a product of conditional probabilities of normalized wavelet coefficients across scales, as shown in Marchand et al. (2022). These conditional probabilities are more similar to Gaussian white noise than the original image distribution, and can thus be sampled more efficiently. On the physics side, this observation is rooted in the renormalization group decomposition in statistical physics (Wilson, 1971), and has been used to estimate physical energies from data (Marchand et al., 2022). In image processing, it relies on statistical observations of wavelet coefficient properties (Wainwright and Simoncelli, 1999). A Wavelet Score-based Generative Model (WSGM) generates normalized wavelet coefficients from coarse to fine scales, as illustrated in Figure 3.1. The conditional distribution of each set of wavelet coefficients, given coarse scale coefficients, is sampled with its own (conditional) SGM. The main result is that a normalization of wavelet coefficients allows fixing the same discretization schedule at all scales. Remarkably, and as opposed to existing algorithms, it implies that the total number of sampling iterations per image pixel does not depend on the image size.

After reviewing score-based generation models, Section 3.2 studies the mathematical properties of its time discretization, with a focus on Gaussian models and multiscale processes. Images and many physical processes are typically non-Gaussian, but do have a singular covariance with long- and short-range correlations. In Section 3.3, we review how to factorize these processes into probability distributions which capture interactions across scales by introducing orthogonal wavelet transforms. We shall prove that it allows considering SGMs with the same time schedule at all scales, independently of the image size. In Section 3.4, we present numerical results on Gaussian distributions, the φ^4 physical model at phase transition, and the CelebA-HQ image

dataset (Karras et al., 2018).

We omit the proofs of the mathematical results of this chapter, which were not done by the author of this manuscript. We refer the reader to the original publication (Guth et al., 2022a).

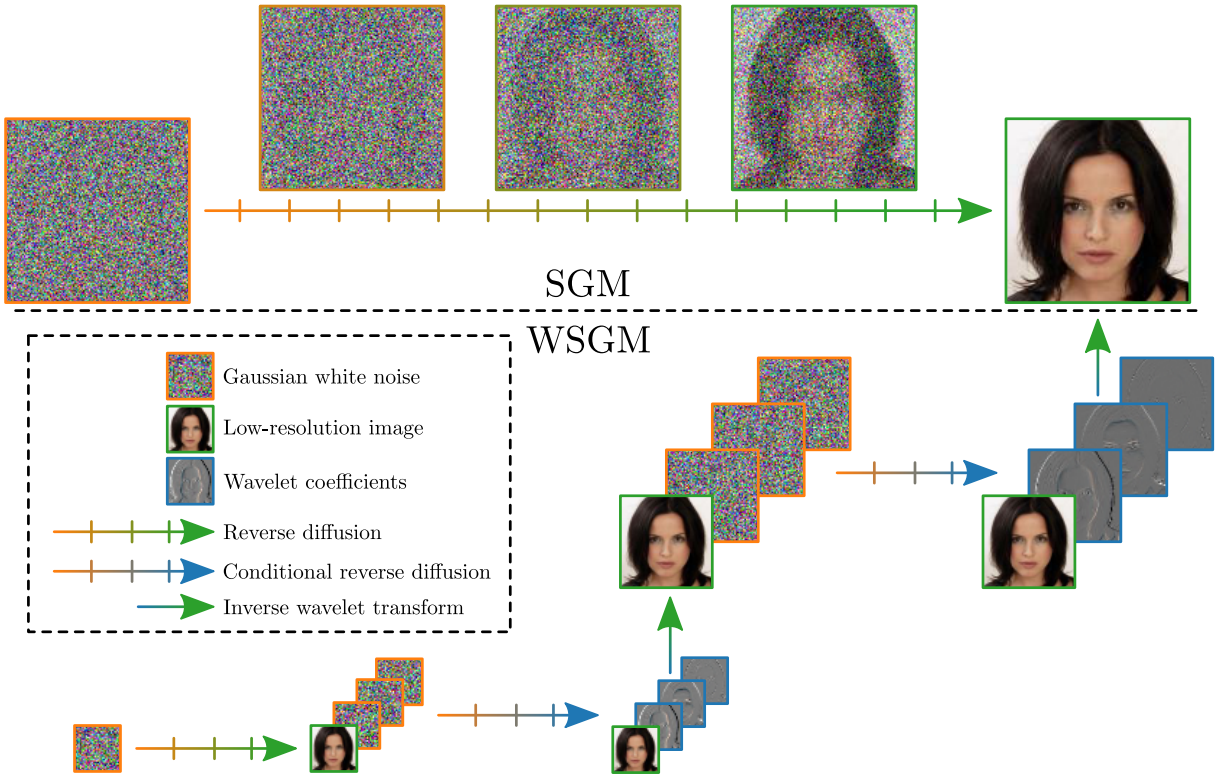

FIGURE 3.1: An SGM generates images by discretizing a reverse diffusion, which progressively transforms white Gaussian noise into a natural image. A WSGM generates increasingly higher-resolution images by discretizing reverse diffusions on wavelet coefficients at each scale. It begins by generating a first low-resolution image. Renormalized wavelet coefficients are then generated conditionally to this low-resolution image. A fast inverse wavelet transform reconstructs a higher-resolution image from these wavelet coefficients. This process is repeated at each scale. The number of steps is the same at each scale, and can be orders of magnitude smaller than for SGM.

3.2 Sampling and discretization of score-based generative models

3.2.1 Score-based generative models

Diffusions and time reversal. A Score-based Generative Model (SGM) (Song and Ermon, 2019; Song et al., 2021b; Ho et al., 2020) progressively maps the distribution of data x into the normal distribution, with a forward Stochastic Differential Equation (SDE) which iteratively adds Gaussian white noise. It is associated with a *noising process* $(x_t)_t$, with x_0 distributed according to the data distribution p, and satisfying

$$\mathrm{d}x_t = -x_t \mathrm{d}t + \sqrt{2}\mathrm{d}w_t, \tag{3.1}$$

where $(w_t)_t$ is a Brownian motion. The solution is an Ornstein-Uhlenbeck process which admits the following representation for any $t \geq 0$:

$$x_t = \mathrm{e}^{-t} x_0 + \sqrt{1 - \mathrm{e}^{-2t}} z, \qquad z \sim \mathcal{N}(0, \mathrm{Id}). \tag{3.2}$$

The process $(x_t)_t$ is therefore an interpolation between a data sample x_0 and Gaussian white noise. The *generative process* inverts (3.1). Under mild assumptions on p (Cattiaux et al., 2021; Haussmann and Pardoux, 1986), for any $T \geq 0$, the reverse-time process x_{T-t} satisfies

$$\mathrm{d}x_{T-t} = \{x_{T-t} + 2\nabla \log p_{T-t}(x_{T-t})\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}w_t, \tag{3.3}$$

where p_t is the probability density of x_t, and $\nabla \log p_t$ is called the *Stein score*. Since x_T is close to a white Gaussian random variable, one can approximately sample from x_T by sampling from the normal distribution. We can generate x_0 from x_T by solving this time-reversed SDE, if we can estimate an accurate approximation of the score $\nabla \log p_t$ at each time t, and if we can discretize the SDE without introducing large errors.

Efficient approximations of the Stein scores are the workhorse of SGM. Hyvärinen and Dayan (2005) show that the score $\nabla \log p_t$ can be approximated with parametric functions s_θ which minimize the so-called implicit score matching loss

$$s_t = \arg\min_\theta \mathbb{E}_{p_t}\left[\frac{1}{2}\|s_\theta(x_t)\|^2 + \mathrm{div}(s_\theta)(x_t)\right], \tag{3.4}$$

or, equivalently, the denoising score matching loss

$$s_t = \arg\min_\theta \mathbb{E}_{p_0,\mathcal{N}(0,\mathrm{Id})}\left[\|s_\theta(\mathrm{e}^{-t}x_0 + \sqrt{1-\mathrm{e}^{-2t}}z) + \frac{z}{\sqrt{1-\mathrm{e}^{-2t}}}\|^2\right]. \tag{3.5}$$

For image generation, s_θ is calculated by a neural network parameterized by θ. In statistical physics problems where the energy can be linearly expanded with coupling parameters, we obtain linear models $s_\theta(x) = \theta^{\mathrm{T}}\nabla U(x)$. This is the case for Gaussian processes where $U(x) = xx^{\mathrm{T}}$; it also applies to non-Gaussian processes, using non-quadratic terms in $U(x)$.

Time discretization of generation. An approximation of the generative process (3.3) is computed by approximating $\nabla \log p_t$ by s_t and discretizing time. It amounts to approximating the time-reversed SDE by a Markov chain which is initialised by $\tilde{x}_T \sim \mathcal{N}(0, \mathrm{Id})$, and computed over times t_k which decrease from $t_N = T$ to $t_0 = 0$, at intervals $\delta_k = t_k - t_{k-1}$:

$$\tilde{x}_{t_{k-1}} = \tilde{x}_{t_k} + \delta_k\{\tilde{x}_{t_k} + 2s_{t_k}(\tilde{x}_{t_k})\} + \sqrt{2\delta_k}z_k, \qquad z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathrm{Id}). \tag{3.6}$$

Ignoring the error due to the score model, the minimum number of time steps is limited by the Lipschitz regularity of the score $\nabla \log p_t$, see De Bortoli et al. (2021, Theorem 1). The overall complexity of this generation is N evaluations of the score $s_t(x)$.

3.2.2 Discretization of SGMs and score regularity

We now study how the regularity of the score $\nabla \log p$ affects the discretization of (3.6). Assuming that the score is known, i.e., that $s_t = \nabla \log p_t$, we prove that for Gaussian processes, the number of time steps to reach a fixed error ε depends on the condition number of its covariance. This result is generalized to non-Gaussian processes by relating this error to the regularity of $\nabla \log p_t$.

Gaussian distributions. Suppose that the data distribution is a Gaussian $p = \mathcal{N}(0, \Sigma)$ with covariance matrix Σ, in dimension d. Let p_t be the distribution of x_t. Using (3.2), we have

$$\nabla \log p_t(x) = -(\mathrm{Id} + (\Sigma - \mathrm{Id})\mathrm{e}^{-2t})^{-1}x. \tag{3.7}$$

Let $\tilde{p}_t$ be the distribution of $\tilde{x}_t$ obtained by the time discretization (3.6). The approximation error between the distribution $\tilde{p}_0$ obtained with the time-reversed SDE and the data distribution p stems from (i) the mismatch between the distributions of x_T and $\tilde{x}_T$, and (ii) the time

discretization. The following theorem relates these two errors to the covariance Σ of x in the particular case of a uniform time sampling at intervals $\delta_k = \delta$. We normalize the signal energy by imposing that $\mathrm{Tr}(\Sigma) = d$, and we write κ the condition number of Σ, which is the ratio between its largest and smallest eigenvalues.

Theorem 3.1. *If the data distribution $p = \mathcal{N}(0, \Sigma)$, the distribution $\tilde{p}_0$ of $\tilde{x}_0$ in (3.6) with a uniform discretization $\delta_k = \delta$ satisfies $\mathrm{KL}\, p\tilde{p}_0 \leq E_T + E_\delta + E_{T,\delta}$, with*

$$E_T = f(\mathrm{e}^{-4T}\, |\mathrm{Tr}((\Sigma - \mathrm{Id})\Sigma))|), \tag{3.8}$$
$$E_\delta = f(\delta |\mathrm{Tr}(\Sigma^{-1} - \Sigma(\Sigma - \mathrm{Id})^{-1} \log(\Sigma)/2 + (\mathrm{Id} - \Sigma^{-1})/3))|), \tag{3.9}$$

where $f(t) = t - \log(1+t)$ and $E_{T,\delta}$ is a higher-order term with $E_{T,\delta} = o(\delta + \mathrm{e}^{-4T})$ when $\delta \to 0$ and $T \to +\infty$. Furthermore, for any $\varepsilon > 0$, there exists $T, \delta \geq 0$ such that

$$(1/d)(E_T + E_\delta) \leq \varepsilon \quad \text{and} \quad N = T/\delta \leq C\varepsilon^{-2}\kappa^3. \tag{3.10}$$

with $C \geq 0$ a universal constant and κ the conditioning number of Σ.

This theorem specifies the dependence of the Kullback-Leibler error on the covariance matrix. It computes an upper bound on the number of time steps $N = T/\delta$ to reach an error ε as a function of the condition number κ of Σ. As expected, it indicates that the number of time steps should increase with the condition number of the covariance. This theorem is proved in a more general case in Guth et al. (2022a, Appendix, S5), which includes the case where p has a non-zero mean. An exact expansion of the Kullback-Leibler divergence is also given.

For stationary processes of images, the covariance eigenvalues are given by the power spectrum, which typically decays like $|\omega|^{-1}$ at a frequency ω. It results that κ is proportional to a power of the image size. Many physical phenomena produce such stationary images with a power spectrum having a power law decay. In these typical cases, the number of time steps must increase with the image size. This is indeed what is observed in numerical SGM experiments, as seen in Section 3.3.

General processes. Theorem 3.1 can be extended to non-Gaussian processes. The number of time steps then depends on the regularity of the score $\nabla \log p_t$.

Theorem 3.2. *Assume that $\nabla \log p_t(x)$ is $\mathscr{C}^2$ in both t and x, and that*

$$\sup_{x,t} \|\nabla^2 \log p_t(x)\| \leq K \quad \text{and} \quad \|\partial_t \nabla \log p_t(x)\| \leq M\, \mathrm{e}^{-\alpha t}\, \|x\|. \tag{3.11}$$

for some $K, M, \alpha > 0$. Then $\|p - \tilde{p}_0\|_{\mathrm{TV}} \leq E_T + E_\delta + E_{T,\delta}$, where

$$E_T = \sqrt{2}\mathrm{e}^{-T}\, \mathrm{KL}(p \| \mathcal{N}(0, \mathrm{Id}))^{1/2}, \tag{3.12}$$
$$E_\delta = 6\sqrt{\delta}\, [1 + \mathbb{E}_p(\|x\|^4)^{1/4}]\, [1 + K + M(1 + 1/(2\alpha)^{1/2})], \tag{3.13}$$

and $E_{\delta,T}$ is a higher order term with $E_{T,\delta} = o(\sqrt{\delta} + \mathrm{e}^{-T})$ when $\delta \to 0$ and $T \to +\infty$.

The proof of Theorem 3.2 is in the original publication (Guth et al., 2022a, Appendix S5) which shows that the result can be strengthened by providing a quantitative upper bound on $\|p - \tilde{p}_0\|_{\mathrm{TV}}$. Theorem 3.2 improves on (De Bortoli et al., 2021, Theorem 1) by proving explicit bounds exhibiting the dependencies on the regularity constants K and M of the score and by eliminating an exponential growth term in T in the upper bound. Theorem 3.2 is much more general but not as tight as Theorem 3.1.

The first error term (3.12) is due to the fact that T is chosen to be finite. The second error term (3.13) controls the error depending upon the discretization time step δ. Since p_t is

obtained from p through a high-dimensional convolution with a Gaussian convolution of variance proportional to t, the regularity of $\nabla \log p_t(x)$ typically increases with t so $\|\nabla^2 \log p_t(x)\|$ and $\|\partial_t \nabla \log p_t(x)\|$ rather decrease when t increases. This qualitatively explains why a *quadratic* discretization schedule with non-uniform time steps $\delta_k \propto k$ are usually chosen in numerical implementations of SGMs (Nichol and Dhariwal, 2021; Song and Ermon, 2020). For simplicity, we focus on the uniform discretization schedule, but our result could be adapted to non-uniform time steps with no major difficulties. This remark also explains that it is mainly the regularity of the score at time $t = 0$ $\nabla \log p$ which determines the error decay (3.13).

While Theorem 3.2 is more general than Theorem 3.1, the Gaussian case provides intuition about the speed of the error decay (3.13) through the value of the constants K and M. If p is Gaussian, then the Hessian $\nabla^2 \log p$ is the negative inverse of the covariance matrix. It is verified in Guth et al. (2022a, Appendix S5) that in this case, the assumptions of Theorem 3.2 are satisfied. Furthermore, the constants K and M, and hence the number of discretization steps, are controlled using the condition number of Σ. We thus conjecture that non-Gaussian processes with an ill-conditioned covariance matrix will require many discretization steps to have a small error. This will be verified numerically. As we now explain, such processes are ubiquitous in physics and natural image datasets.

Multiscale processes. Most images have variations on a wide range of scales. They require to use many time steps to sample using an SGM, because their score is not well-conditioned. This is also true for a wide range of phenomena encountered in physics, biology, or economics (Kolmogorov, 1962; Mandelbrot, 1983). We define a *multiscale process* as a stationary process whose power spectrum has a power law decay. The stationarity implies that its covariance is diagonalized in a Fourier basis. Its eigenvalues, which then coincide with its power spectrum, have a power law decay defined by

$$P(\omega) \sim (\xi^\eta + |\omega|^\eta)^{-1}, \tag{3.14}$$

where $\eta > 0$ and $2\pi/\xi$ is the maximum correlation length. Physical processes near phase transitions have such a power-law decay, but it is also the case of many disordered systems such as fluid and gas turbulence. Natural images also typically define stationary processes. Their power spectrum satisfy this property with $\eta = 2$ and $2\pi/\xi \approx L$ for images of size $L \times L$. To efficiently synthesize images and more general multiscale signals, we must eliminate the ill-conditioning properties of the score. This is done by applying a wavelet transform.

3.3 Wavelet score-based generative models

The numerical complexity of the SGM algorithm depends on the number of time steps, which itself depends upon the regularity of the score. We show that an important acceleration is obtained by factorizing the data distribution into normalized wavelet conditional probability distributions, which are closer to a white Gaussian distribution, and so whose score is better-conditioned.

3.3.1 Wavelet whitening and cascaded SGMs

Normalized orthogonal wavelet coefficients. Let x be the input signal of width L and dimension $d = L^n$, with $n = 2$ for images. We write x_j its low-frequency approximation subsampled at intervals 2^j, of size $(2^{-j}L)^n$, with $x_0 = x$. At each scale $2^{j-1} \geq 1$, a fast wavelet orthogonal transform decomposes x_{j-1} into $(\bar{x}_j, x_j)$ where $\bar{x}_j$ are the wavelet coefficient which carries the higher frequency information over $2^n - 1$ signals of size $(2^{-j}L)^n$ (Mallat, 1989). They are calculated with convolutional and subsampling operators G and $\bar{G}$ specified in Appendix B.2:

$$x_j = \gamma_j^{-1} G\, x_{j-1} \quad \text{and} \quad \bar{x}_j = \gamma_j^{-1} \bar{G}\, x_{j-1}\ . \tag{3.15}$$

The normalization factor γ_j guarantees that $\mathbb{E}[\|\bar{x}_j\|^2] = (2^n - 1)(2^{-j}L)^n$. We consider wavelet orthonormal filters where $(G, \bar{G})$ is a unitary operator, i.e.,

$$\bar{G}G^{\mathrm{T}} = G\bar{G}^{\mathrm{T}} = 0 \quad \text{and} \quad G^{\mathrm{T}}G + \bar{G}^{\mathrm{T}}\bar{G} = \mathrm{Id}\,.$$

It results that x_{j-1} is recovered from $(\bar{x}_j, x_j)$ with

$$x_{j-1} = \gamma_j\, G^{\mathrm{T}} x_j + \gamma_j\, \bar{G}^{\mathrm{T}} \bar{x}_j. \tag{3.16}$$

The wavelet transform is computed over $J \approx \log_2 L$ scales by iterating J times on (3.15). The last x_J has a size $(2^{-J}L)^n \approx 1$. Appendix B.2 contains a more detailed introduction to the wavelet transform. The choice of wavelet filters G and $\bar{G}$ specifies the properties of the wavelet transform and the number of vanishing moments of the wavelet.

Renormalized probability distribution. A conditional wavelet renormalization factorizes the distribution $p(x)$ of signals x into conditional probabilities over wavelet coefficients:

$$p(x) = \alpha \prod_{j=1}^{J} \bar{p}_j(\bar{x}_j|x_j)\, p_J(x_J)\ . \tag{3.17}$$

where α (the Jacobian) depends upon all γ_j.

Although $p(x)$ is typically highly non-Gaussian, the factorization (3.17) involves distributions that are closer to Gaussians. The largest scale distribution p_J is usually close to a Gaussian when the image has independent structures, because x_J is an averaging of x over large domains of size 2^J. In images, the wavelet coefficients $\bar{x}_j$ are usually sparse and thus have a highly non-Gaussian distribution; however, it has been observed (Wainwright and Simoncelli, 1999) that their conditional distributions $\bar{p}_j(\bar{x}_j|x_j)$ become much more Gaussian, due to dependencies of wavelet coefficients across scales. Furthermore, because of the renormalization, the normalized wavelet coefficients $\bar{x}_j$ have a white spectrum, as opposed to a power-law decay for x_j, which implies they are closer to a white Gaussian distribution. In statistical physics, the analysis of high frequencies conditioned by lower frequencies have been studied in Wilson (1983). More recently, normalized wavelet factorizations (3.17) have been introduced in physics to implement renormalization group calculations, and model probability distributions with maximum likelihood estimators near phase transitions (Marchand et al., 2022).

Wavelet score-based generative model. Instead of computing a Score-based Generative Model (SGM) of the distribution $p(x)$, a Wavelet Score-based Generative Model (WSGM) applies an SGM at the coarsest scale $p_J(x_J)$ and then on each conditional distribution $\bar{p}_j(\bar{x}_j|x_j)$ for $j \leq J$. It is thus a cascaded SGM, similarly to Ho et al. (2022); Saharia et al. (2021), but calculated on $\bar{p}_j(\bar{x}_j|x_j)$ instead of $p_j(x_{j-1}|x_j)$. The normalization of wavelet coefficients $\bar{x}_j$ effectively produces a whitening which can considerably accelerate the algorithm by reducing the number of time steps. This is not possible on x_{j-1} because its covariance is ill-conditioned. It will be proved for Gaussian processes.

A forward noising process is computed on each $\bar{x}_j$ for $j \leq J$ and x_J:

$$\mathrm{d}\bar{x}_{j,t} = -\bar{x}_{j,t}\,\mathrm{d}t + \sqrt{2}\mathrm{d}\bar{w}_{j,t} \quad \text{and} \quad \mathrm{d}x_{J,t} = -x_{J,t}\,\mathrm{d}t + \sqrt{2}\mathrm{d}w_{J,t}, \tag{3.18}$$

where the $\bar{w}_{j,t}, w_{J,t}$ are Brownian motions. Since $\bar{x}_j$ is nearly white and has Gaussian properties, this diffusion converges much more quickly than if applied directly on x. Using (3.4) or (3.5), we compute a score function $s_{J,t}(x_{J,t})$ which approximates the score $\nabla \log p_{J,t}(x_{J,t})$. For each $j \leq J$ we also compute the conditional score $\bar{s}_{j,t}(\bar{x}_{j,t}|x_j)$ which approximates $\nabla \log \bar{p}_{j,t}(\bar{x}_{j,t}|x_j)$.

The inverse generative process is computed from coarse to fine scales as follows. At the largest scale 2^J, we sample the low-dimensional x_J by discretizing the inverse SDE. Similarly to (3.6), the generative process is given by

$$x_{J,t_{k+1}} = x_{J,t_k} + \delta_k\{x_{J,t_k} + 2s_{J,t_k}(x_{J,t_k})\} + \sqrt{2\delta_k}z_{J,k}, \qquad z_{J,k} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathrm{Id}). \tag{3.19}$$

For j going from J to 1, we then generate the wavelet coefficients $\bar{x}_j$ conditionally to the previously calculated x_j, by keeping the same time discretization schedule at all scales:

$$\bar{x}_{j,t_{k+1}} = \bar{x}_{j,t_k} + \delta_k\{\bar{x}_{j,t_k} + 2\bar{s}_{j,t_k}(\bar{x}_{j,t_k}|x_j)\} + \sqrt{2\delta_k}\, z_{j,k}, \qquad z_{j,k} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathrm{Id}). \tag{3.20}$$

The inverse wavelet transform then approximately computes a sample of x_{j-1} from $(\bar{x}_{j,0}, x_j)$:

$$\tilde{x}_{j-1} = \gamma_j\, G^{\mathrm{T}} x_j + \gamma_j\, \bar{G}^{\mathrm{T}} \bar{x}_{j,0}. \tag{3.21}$$

The generative process is illustrated in Figure 3.1 and its pseudocode is given in Algorithm B.1 in Appendix B.1. The appendix also verifies that if x is of size d then the numerical complexity of the generation is $O(Nd)$, where N is the number of time steps, which is the same at each scale. For multiscale processes, we shall see that the number of time steps N does not depend upon d to reach a fixed error measured with a KL divergence.

Related work. Multi-scale representations, based on wavelets or not, have been incorporated in many generative modeling approaches in order to increase generation quality and sampling efficiency. Specifically, they have been shown to improve results for auto-encoders (Chen et al., 2018), GANs (Gal et al., 2021) and normalizing flows (Li, 2021). Closer in spirit to our work, Yu et al. (2020) introduces Wavelet Flow, a normalizing flow with a cascade of layers generating wavelet coefficients conditionally on lower-scales, then aggregating them with an inverse wavelet transform. This method yields training time acceleration and high-resolution (1024×1024) generation.

WSGM is closely related to other cascading diffusion algorithms, such as the ones introduced in Ho et al. (2022); Saharia et al. (2021); Dhariwal and Nichol (2021). The main difference lies in that earlier works on cascaded SGMs do not model the *wavelet coefficients* $\{\bar{x}_j\}_{j=1}^J$ but the *low-frequency* coefficients $\{x_j\}_{j=1}^J$. As a result, cascaded models do not explicitly exploit the whitening properties of the wavelet transform, nor the fact that conditional wavelet distributions are often nearly Gaussian, and the mechanisms behind the acceleration remain implicit. We also point out the recent work of Jing et al. (2022) which, while not using the cascading framework, drop subspaces from the noising process at different times. This allows using only one SDE to sample approximately from the data distribution. However, the reconstruction is still computed with respect to $\{x_j\}_{j=1}^J$ instead of the wavelet coefficients.

Finally, we highlight that our work could be combined with other acceleration techniques such as the ones of Jolicoeur-Martineau et al. (2021); Liu et al. (2022a); Zhang and Chen (2022); San-Roman et al. (2021); Nachmani et al. (2021); Song et al. (2020); Ho et al. (2020); Kong and Ping (2021); Luhman and Luhman (2021); Salimans and Ho (2022); Xiao et al. (2021) in order to improve the empirical results of WSGM.

3.3.2 Discretization and accuracy for Gaussian processes

We now illustrate Theorem 3.1 and the effectiveness of WSGM on Gaussian multiscale processes. We use the whitening properties of the wavelet transform to show that the time complexity required in order to reach a given error is linear in the image dimension.

The following result proves that the normalization of wavelet coefficients performs a pre-conditioning of the covariance, whose eigenvalues then remain of the order of 1. This is a consequence of a theorem proved by Meyer (1992) on the representation of classes of singular operators in wavelet bases. As a result, the number of iterations $N = T/\delta$ required to reach an error ε is independent of the dimension.

Theorem 3.3. *Let x be a Gaussian stationary process of power spectrum $P(\omega) = c\,(\xi^\eta + |\omega|^\eta)^{-1}$ with $\eta > 0$ and $\xi > 0$. If the wavelet has a compact support, $q \geq \eta$ vanishing moments and is*

$\mathscr{C}^q$, then the first-order terms E_T and E_δ in the sampling error of WSGM $\mathrm{KL}\, p\tilde{p}_0$ *are such that for any $\varepsilon > 0$, there exists $C > 0$ such that for any δ, T,*

$$(1/d)(E_T + E_\delta) \leq \varepsilon \quad \textit{and} \quad N = T/\delta \leq C\varepsilon^{-2}. \tag{3.22}$$

To prove this result, we show that the conditioning number of the covariance matrix of the renormalized wavelet coefficients does not depend on the dimension, by using Sobolev norm equivalences (Jaffard, 1992; Meyer, 1992). We conclude upon combining this result, the cascading property of the Kullback-Leibler divergence and an extension of Theorem 3.1 to the setting with non-zero mean. The detailed proof is in the original publication (Guth et al., 2022a, Appendix S6).

Numerical results. We illustrate Theorem 3.3 on a Gaussian field x, whose power spectrum P has a power law decay (3.14). In Figure 3.2, we display the sup-norm between P and the power spectrum $\hat{P}$ of the samples obtained using either vanilla SGM or WSGM with uniform stepsize $\delta_k = \delta$. In the case of vanilla SGM, the number $N(\varepsilon)$ of time steps needed to reach a small error $\|P - \hat{P}\| = \varepsilon$ increases with the size of the image L (Fig. 3.2, right). Equation (3.10) suggests that $N(\varepsilon)$ scales like a power of the conditioning number κ of Σ, which is for multiscale Gaussian processes $\kappa \sim L^\eta$, for images of size $L \times L$. In the WSGM case, we sample from the conditional distributions $\bar{p}_j$ of wavelet coefficients $\bar{x}_j$ given low frequencies x_j. At a scale j, the conditioning numbers $\bar{\kappa}_j$ of the conditional covariance become dimension-independent, removing the dependency of $N(\varepsilon)$ on the image size L as suggested by (3.22).

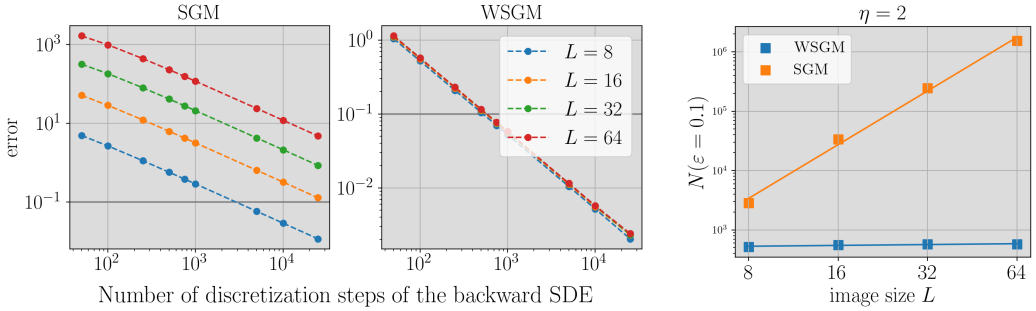

Figure 3.2: Left and middle: *evolution of the error on the estimated covariance matrix using either SGM or WSGM w.r.t. the number of stepsizes used in the model ($T = 10$ is fixed).* Right: *number $N(\varepsilon)$ of discretization steps required to reach a given error $\varepsilon = 0.1$ using either SGM or WSGM.*

3.4 Acceleration with WSGM: numerical results

For multiscale Gaussian processes, we proved that with WSGMs, the number of time steps $N(\varepsilon)$ to reach a fixed error ε does not depend on the signal size, as opposed to SGMs. This section shows that this result applies to non-Gaussian multiscale processes. We consider a physical process near a phase transition and images from the CelebA-HQ database (Karras et al., 2018).

3.4.1 Physical processes with scalar potentials

Gaussian stationary processes are maximum entropy processes conditioned by second order moments defined by a circulant matrix. More complex physical processes are modeled by imposing a constraint on their marginal distribution, with a so-called scalar potential. The marginal distribution of x is the probability distribution of an image pixel $x(u)$, which does not depend upon u if x is stationary. Maximum entropy processes conditioned by second order moments and

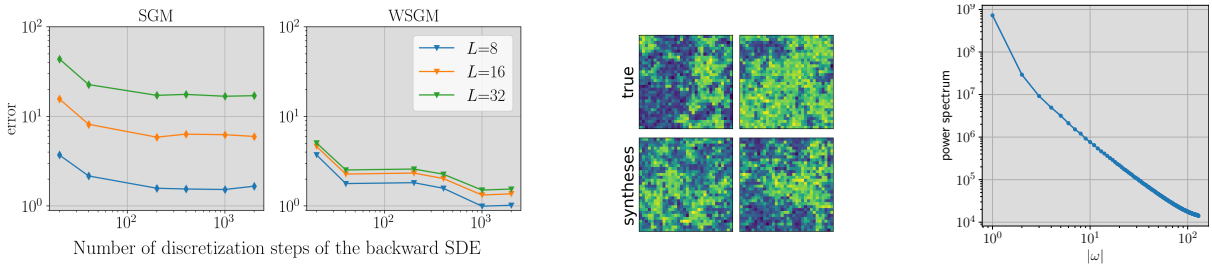

FIGURE 3.3: Left: *error between ground-truth φ^4 datasets in various dimensions L, and the synthetized datasets with SGM and WSGM, for various number of discretization steps.* Middle: *realizations of φ^4 (top) and WSGM samples (bottom).* Right: *power spectrum of φ^4 for $L = 256$.*

marginal distributions have a probability density which is a Gibbs distribution $p(x) = Z^{-1}\,\mathrm{e}^{-E(x)}$ with

$$E(x) = \tfrac{1}{2} x^{\mathrm{T}} C x + \textstyle\sum_u V(x(u)) \quad , \tag{3.23}$$

where C is a circulant matrix and $V : \mathbb{R} \to \mathbb{R}$ is a scalar potential. Appendix B.4 explains how to parameterize V as a linear combination of a family of fixed elementary functions. The φ^4 model is a particular example where $C = -\Delta$ is the negative Laplacian and V is a fourth-order polynomial, adjusted in order to impose that $x(u) \approx \pm 1$ with high probability. For so-called critical values of these parameters, the resulting process becomes multiscale with long range interactions and a power law spectrum, see Figure 3.3-(c).

We train SGMs and WSGMs on critical φ^4 processes of different sizes; for the score model s_θ, we use a simple linear parameterization detailed in Appendix B.4. To evaluate the quality of the generated samples, it is sufficient to verify that these samples have the same second order moment and marginals as φ^4. We define the error metric as the sum of the L^2 error on the power spectrum and the total-variation distance between marginal distributions. Figure 3.3-(a) shows the decay of this error as a function of the number of time steps used in an SGM and WSGM with a uniform discretization. With vanilla SGM, the loss has a strong dependency in L, but becomes almost independent of L for WSGM. This empirically verifies the claim that an ill-conditioned covariance matrix leads to slow sampling of SGM, and that WSGM is unaffected by this issue by working with the conditional distributions of normalized wavelet coefficients.

3.4.2 Scale-wise time reduction in natural images

Images are highly non-Gaussian multiscale processes whose power spectrum has a power law decay. We now show that WSGM also provides an acceleration over SGM in this case, by being independent of the image size.

We focus on the CelebA-HQ dataset (Liu et al., 2015) at the 128×128 resolution. Its power spectrum has a power law decay, as shown in Figure 3.4, and it thus suffers from ill-conditioning, even though it is a non-stationary process. We compare SGM (Ho et al., 2020) samples at the 128×128 resolution with WSGM samples which start from the 32×32 resolution. Though smaller, the 32×32 resolution still suffers from a power law decay of its spectrum over several orders of magnitude. The reason why we limit this coarsest resolution is because border effects become dominant at lower image sizes. To simplify the handling of border conditions, we use Haar wavelets.

Following Nichol and Dhariwal (2021), the global scores $s_\theta(x)$ are parameterized by a neural network with a U-Net architecture. It has 3 residual blocks at each scale, and includes multi-head attention layers at lower scales. The conditional scores $s_\theta(\bar{x}_j | x_j)$ are parameterized in the same way, and the conditioning on the low frequencies x_j is done with a simple input concatenation along channels (Nichol and Dhariwal, 2021; Saharia et al., 2021). The details of the architecture

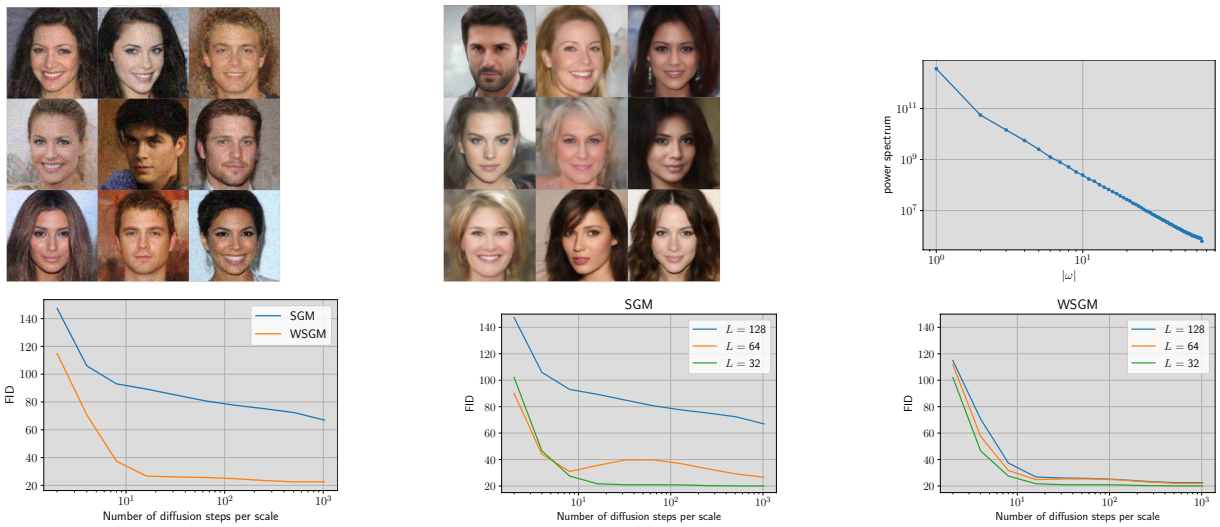

Figure 3.4: Top. *(a): Generations from SGM with 16 discretization steps. (b): Generations from WSGM with 16 discretization steps at each scale. (c): Power spectrum of CelebA-HQ.* Bottom. *(a): Evolution of the FID w.r.t. the number of diffusion steps for SGM and WSGM with $L = 128$. (b): Evolution of the FID w.r.t. the number of diffusion steps for SGM at several image sizes L. (c) Evolution of the FID w.r.t. the number of diffusion steps for WSGM at several image sizes L.*

are in Appendix B.5. We use a uniform discretization of the backward SDE to stay in the setting of Theorem 3.2, and show that WSGM still obtains satisfactory results in this case.

The generation results are given in Figure 3.4. With the same computational budget of 16 discretizations steps at the largest scale (iterations at smaller scales having a negligible cost due to the exponential decrease in image size), WSGM achieves a much better perceptual generation quality. Notably, SGM generates noisy images due to discretization errors. This is confirmed quantitatively with the Fréchet Inception Distance (FID) (Heusel et al., 2017). The FID of the WSGM generations decreases with the number of steps, until it plateaus. This plateau is reached with at least 2 orders of magnitude less steps for WSGM than SGM. This number of steps is also independent of the image size for WSGM, thus confirming the intuition given in the Gaussian case by Theorems 3.1 and 3.3. Our results confirm that vanilla SGM on a wide range of multiscale processes, including natural images, suffers from ill-conditioning, in the sense that the number of discretization steps grows with the image size. WSGM, on the contrary, leads to uniform discretization schemes whose number of steps at each scale does not depend on the image size.

We also stress that there exists many techniques (Kadkhodaie and Simoncelli, 2021; Jolicoeur-Martineau et al., 2021; Liu et al., 2022a; Zhang and Chen, 2022; San-Roman et al., 2021; Nachmani et al., 2021; Song et al., 2020; Kong and Ping, 2021; Ho et al., 2020; Luhman and Luhman, 2021; Salimans and Ho, 2022; Xiao et al., 2021) to accelerate the sampling of vanilla SGMs, with sometimes better FID-time complexity tradeoff curves. Notably, the FID plateaus at a relatively high value of 20 because the coarsest resolution 32×32 is still ill-conditioned, and thus requires thousands of steps with a non-uniform discretization schedule to achieve FIDs less than 10 with vanilla SGMs (Nichol and Dhariwal, 2021). Such improvements (including proper handling of border conditions) are beyond of the scope of this chapter. The contribution of WSGM is rather to show the reason behind this sampling inefficiency and mathematically prove in the Gaussian setting that wavelet decompositions of the probability distribution allows solving this problem. Extending this theoretical result to a wider class of non-Gaussian multiscale processes, and combining WSGM with other sampling accelerations, are interesting research directions.

3.5 Discussion

This chapter introduces a Wavelet Score-based Generative Model (WSGM) which applies an SGM to normalized wavelet coefficients conditioned by lower frequencies. We prove that the number of steps in SGMs is controlled by the regularity of the score of the target distribution. For multiscale processes such as images, it requires a considerable number of time steps to achieve a good accuracy, which increases quickly with the image size. We show that a WSGM eliminates ill-conditioning issues by normalizing wavelet coefficients. As a result, the number of steps in WSGM does not increase with the image size. We illustrated our results on Gaussian distributions, physical processes, and image datasets.

One of the main limitations of the work presented in this chapter is that the theoretical analysis of the advantages brought by the wavelet conditional factorization is limited to the Gaussian case. For natural images, we observe empirically a reduced sampling complexity, which we motivated from a well-conditioned *conditional* covariance and thus a more regular conditional score, but it is not clear if this argument can be made more precise. This observation is complemented by the results of Chapter 2, which show that the wavelet conditional factorization reveals the much stronger property of conditional log-concavity for multiscale physical fields. Wavelet score-based generative models of natural images are further studied in Chapter 4, which focuses on the estimation of the conditional scores.

Chapter

4 Multiscale Local Conditional Models of Images

Chapter content

4.1 Introduction 57
4.2 Markov wavelet conditional models 59
4.3 Score-based markov wavelet conditional models 60
4.4 Markov wavelet conditional denoising 62
4.5 Markov wavelet conditional super-resolution and synthesis 64
4.6 Discussion 66

We have shown in Chapter 3 that a multiscale conditional factorization of image distributions can reduce the sampling complexity of score-based diffusion models from quadratic (Chen et al., 2022a) to (empirically) linear in the dimension. However, the major theoretical challenge remains to understand how deep neural networks manage to learn the scores of the probability distribution and thereby capture complex global statistical structure, apparently without suffering from the curse of dimensionality.

In this chapter, we show that the multiscale conditional factorization studied in Chapters 2 and 3 allows reducing the dimensionality of the score matching task. This is achieved by assuming a stationary local Markov model for wavelet coefficients conditioned on coarser-scale coefficients, similarly to the one introduced by Marchand et al. (2022) and used in Chapter 2. We instantiate this model using convolutional neural networks (CNNs) with local receptive fields, which enforce both the stationarity and Markov properties. Global structures are captured using a CNN with receptive fields covering the entire (but small) low-pass image. We test this model on a dataset of face images, which are highly non-stationary and contain large-scale geometric structures. Remarkably, denoising, super-resolution, and image synthesis results all demonstrate that these structures can be captured with significantly smaller conditioning neighborhoods than required by a Markov model implemented in the pixel domain. Our results show that score estimation for large complex images can be reduced to low-dimensional Markov conditional models across scales, partially alleviating the curse of dimensionality.

This chapter is adapted from the following publication: Zahra Kadkhodaie, Florentin Guth, Stéphane Mallat, and Eero P Simoncelli. Learning multi-scale local conditional probability models of images. In *International Conference on Learning Representations*, 2023.

4.1 Introduction

Deep neural networks (DNNs) have produced dramatic advances in synthesizing complex images and solving inverse problems, all of which rely (at least implicitly) on prior probability models. Of particular note is the recent development of "diffusion methods" (Sohl-Dickstein et al., 2015), in which a network trained for image denoising is incorporated into an iterative algorithm to draw samples from the prior (Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021b), or

to solve inverse problems by sampling from the posterior (Kadkhodaie and Simoncelli, 2021; Cohen et al., 2021; Kawar et al., 2021; Daras et al., 2022). The prior in these procedures is implicitly defined by the learned denoising function, which depends on the prior through the score (the gradient of the log density). But density or score estimation is notoriously difficult for high-dimensional signals because of the curse of dimensionality: worst-case data requirements grow exponentially with the data dimension. How do neural network models manage to avoid this curse?

Traditionally, density estimation is made tractable by assuming simple low-dimensional models, or structural properties that allow factorization into products of such models. For example, the classical Gaussian spectral model for images or sounds rests on an assumption of translation-invariance (stationarity), which guarantees factorization in the Fourier domain. Markov random fields (Geman and Geman, 1984) assume localized conditional dependencies, which guarantees that the density can be factorized into terms acting on local, typically overlapping neighborhoods (Clifford and Hammersley, 1971). In the context of images, these models have been effective in capturing local properties, but are not sufficiently powerful to capture long-range dependencies. Multiscale image decompositions offered a mathematical and algorithmic framework better suited for the structural properties of images (Burt and Adelson, 1983; Mallat, 2008). The multiscale representation facilitates handling of larger structures, and local (Markov) models have captured these probabilistically (e.g., Chambolle et al. (1998); Malfait and Roose (1997); Crouse et al. (1998); Buccigrossi and Simoncelli (1999); Paget and Longstaff (1998); Mihçak et al. (1999); Wainwright et al. (2001b); Şendur and Selesnick (2002); Portilla et al. (2003); Cui and Wang (2005); Lyu and Simoncelli (2009)). Recent work, inspired by renormalization group theory in physics, has shown that probability distributions with long-range dependencies can be factorized as a product of Markov *conditional* probabilities over wavelet coefficients (Marchand et al., 2022). Although the performance of these models is eclipsed by recent DNN models, the concepts on which they rest—stationarity, locality and multiscale conditioning—are still of fundamental importance. Here, we use these tools to constrain and study a score-based diffusion model.

A number of recent DNN image synthesis methods—including variational auto-encoders (Chen et al., 2018), generative adversarial networks (Gal et al., 2021) normalizing flow models (Yu et al., 2020; Li, 2021)), and diffusion models (Ho et al., 2022)—use coarse-to-fine strategies, generating a sequence of images of increasing resolution, each seeded by its predecessor. With the exception of the last, these do not make explicit the underlying conditional densities, and none impose locality restrictions on their computation. On the contrary, the stage-wise conditional sampling is typically accomplished with huge DNNs (up to billions of parameters), with global receptive fields.

Here, we develop a low-dimensional probability model for images decomposed into multiscale wavelet sub-bands. Following the renormalization group approach, the image probability distribution is factorized as a product of conditional probabilities of its wavelet coefficients conditioned by coarser scale coefficients. We assume that these conditional probabilities are local and stationary, and hence can be captured with low-dimensional Markov models. Each conditional score can thus be estimated with a conditional CNN (cCNN) with a small receptive field (RF). The score of the coarse-scale low-pass band (a low-resolution version of the image) is modeled using a CNN with a global RF, enabling representation of large-scale image structures and organization. We test this model on a dataset of face images, which present a challenging example because of their global geometric structure. Using a coarse-to-fine anti-diffusion strategy for drawing samples from the posterior (Kadkhodaie and Simoncelli, 2021), we evaluate the model on denoising, super-resolution, and synthesis, and show that locality and stationarity assumptions hold for conditional RF sizes as small as 9×9 without harming performance. In comparison, the performance of CNNs restricted to a fixed RF size in the pixel domain dramatically degrades when the RF is reduced to such sizes. Thus, high-dimensional score estimation

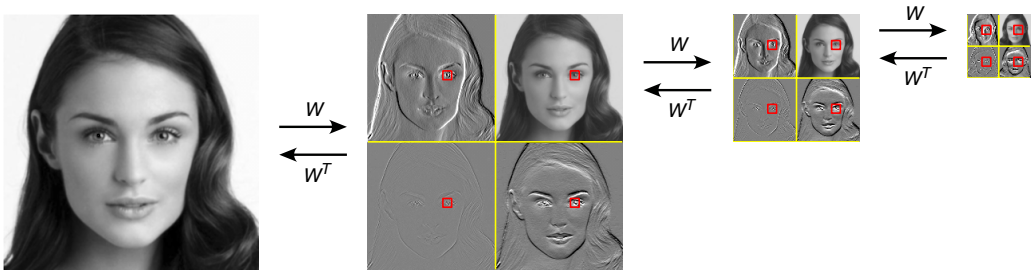

FIGURE 4.1: Markov wavelet conditional model structure. At each scale j, an orthogonal wavelet transform W decomposes an image x_{j-1} into three wavelet channels, $\bar{x}_j$, containing vertical, horizontal, and diagonal details, and a low-pass channel x_j containing a coarse approximation of the image, all subsampled by a factor of two. At each scale j, we assume a Markov wavelet conditional model, in which the probability distribution of any wavelet coefficient of $\bar{x}_j$ (here, centered on the left eye), conditioned on values of x_j and $\bar{x}_j$ in a local spatial neighborhood (red squares), is independent of all coefficients of $\bar{x}_j$ outside this neighborhood.

for images can be reduced to low-dimensional Markov conditional models, alleviating the curse of dimensionality.

4.2 Markov wavelet conditional models

Images are high-dimensional vectors. Estimating an image probability distribution or its score therefore suffer from the curse of dimensionality, unless one limits the estimation to a relatively low-dimensional model class. This section introduces such a model class as a product of Markov conditional probabilities over multiscale wavelet coefficients.

Markov random fields (Dobrushin, 1968; Sherrington and Kirkpatrick, 1975) define low-dimensional models by assuming that the probability distribution has local conditional dependencies over a graph, which is known a priori. One can then factorize the probability density into a product of conditional probabilities, each defined over a small number of variables (Clifford and Hammersley, 1971). Markov random fields have been used to model stationary texture images, with conditional dependencies within small spatial regions of the pixel lattice. At a location u, such a Markov model assumes that the pixel value $x(u)$, conditioned on pixel values $x(v)$ for v in a neighborhood of u, is independent from all pixels outside this neighborhood. Beyond stationary textures, however, the chaining of short-range dependencies in pixel domain has proven insufficient to capture the complexity of long-range geometrical structures. Many variants of Markov models have been proposed (e.g., Geman and Geman (1984); Malfait and Roose (1997); Cui and Wang (2005)), but none have demonstrated performance comparable to recent deep networks while retaining a local dependency structure.

Based on the renormalization group approach in statistical physics (Wilson, 1971), new probability models are introduced in Marchand et al. (2022), structured as a product of probabilities of wavelet coefficients conditioned on coarser-scale values, with spatially local dependencies. These Markov conditional models have been applied to ergodic stationary physical fields, with simple conditional Gibbs energies that are parameterized linearly. Here, we generalize such models by parameterizing conditional Gibbs energy gradients with deep conditional convolutional neural networks having a local RF. This yields a class of Markov wavelet conditional models that can generate complex structured images, while explicitly relying on local dependencies to reduce the model dimensionality.

An orthonormal wavelet transform uses a convolutional and subsampling operator W defined with conjugate mirror filters (Mallat, 2008), to iteratively compute wavelet coefficients (see Figure 4.1). Let x_0 be an image of $N \times N$ pixels. For each scale $j > 1$, the operator W

decomposes x_{j-1} into

$$Wx_{j-1} = (\bar{x}_j, x_j),$$

where x_j is a lower-resolution image and $\bar{x}_j$ is an array of three wavelet coefficient images, each with dimensions $N/2^j \times N/2^j$, as illustrated in Figure 4.1. The inverse wavelet transform iteratively computes $x_{j-1} = W^T(\bar{x}_j, x_j)$.

We now introduce the wavelet conditional factorization of probability models. Since W is orthogonal, the probability density of x_{j-1} is also the joint density of $(x_j, \bar{x}_j)$. It can be factorized by conditioning on x_j:

$$p(x_{j-1}) = p(x_j, \bar{x}_j) = p(x_j)p(\bar{x}_j|x_j).$$

This is performed J times, so that the lowest resolution image x_J is small enough, which yields

$$p(x) = p(x_J) \prod_{j=1}^{J} p(\bar{x}_j|x_j). \tag{4.1}$$

The conditional distributions $p(\bar{x}_j|x_j)$ specify the dependencies of image details at scale j conditioned on the coarser scale values, and may be expressed in terms of a conditional Gibbs energy

$$p(\bar{x}_j|x_j) = Z_j(x_j)^{-1} \, \mathrm{e}^{-E_j(\bar{x}_j|x_j)}, \tag{4.2}$$

where $Z_j(x_j)$ is the normalization constant for each x_j. The conditional Gibbs energies (4.2) have been used in the wavelet conditional renormalization group approach to obtain a stable parameterization of the probability model even at critical phase transitions, when the parameterization of the global Gibbs energy becomes singular (Marchand et al., 2022).

Local wavelet conditional renormalization group models (Marchand et al., 2022) further impose that $p(\bar{x}_j|x_j)$ is a conditional Markov random field. That is, the probability distribution of a wavelet coefficient of $\bar{x}_j$ conditioned on values of x_j and $\bar{x}_j$ in a restricted spatial neighborhood is independent of all coefficients of $\bar{x}_j$ and $\bar{x}$ outside this neighborhood (see Figure 4.1). The Hammersley-Clifford theorem states that this Markov property is equivalent to imposing that E_j can be written as a sum of potentials, which only depends upon values of $\bar{x}_j$ and x_j over local cliques (Clifford and Hammersley, 1971). This decomposition substantially alleviates the curse of dimensionality, since one only needs to estimate potentials over neighborhoods of a fixed size which does not grow with the image size. To model ergodic stationary physical fields, the local potentials of the Gibbs energy E_j have been parameterized linearly using physical models Marchand et al. (2022).

We generalize Markov wavelet conditional models by parameterizing the conditional score with a conditional CNN (cCNN) having small receptive fields (RFs):

$$-\nabla_{\bar{x}_j} \log p(\bar{x}_j|x_j) = \nabla_{\bar{x}_j} E_j(\bar{x}_j|x_j). \tag{4.3}$$

Computing the score (4.3) is equivalent to specifying the Gibbs model (4.2) without calculating the normalization constants $Z_j(x_j)$, since these are not needed for noise removal, super-resolution or image synthesis applications.

4.3 Score-based markov wavelet conditional models

Score-based diffusion models have produced impressive image generation results (e.g., Song et al. (2021b); Ho et al. (2022); Rombach et al. (2022); Saharia et al. (2022); Ramesh et al. (2022)). To capture large-scale properties, however, these networks require RFs that encompass the entire image. Our score-based wavelet conditional model leverages the Markov assumption to compute

the score using cCNNs with small receptive fields, offering a low-dimensional parameterization of the image distribution while retaining long-range geometric structures.

Let $y = x + z$ be a noisy observation of a clean image $x \in \mathbb{R}^{N\times N}$ drawn from $p(x)$, with $z \sim \mathcal{N}(0, \sigma^2\mathrm{Id})$ a sample of Gaussian white noise. The minimum mean squared error (MMSE) estimate of the true image is well-known to be the conditional mean of the posterior

$$\hat{x}(y) = \int x p(x|y)\mathrm{d}x. \tag{4.4}$$

This integral can be re-expressed in terms of the score

$$\hat{x}(y) = y + \sigma^2 \nabla_y \log p(y). \tag{4.5}$$

This remarkable result, published in Miyasawa (1961), exposes a direct and explicit relationship between the score of probability distributions and denoising (we reproduce the proof in Appendix C.2 for completeness). Note that the relevant density is not the image distribution, $p(x)$, but the *noisy observation density* $p(y)$. This density converges to $p(x)$ as the noise variance σ^2 goes to zero.

Given this relationship, the score can be approximated with a parametric mapping optimized to estimate the denoised image, $f(y) \approx \hat{x}(y)$. Specifically, we implement this mapping with a CNN, and optimize its parameters by minimizing the denoising squared error $||f(y) - x||^2$ over a large training set of images and their noise-corrupted counterparts. Given eq. (4.5), the denoising residual, $f(y) - y$, provides an approximation of the variance-weighted score, $\sigma^2 \nabla_y \log p(y)$. Also known as denoising score matching (Vincent, 2011), such denoiser-estimated score functions have been used in iterative algorithms for drawing samples from the density (Song et al., 2021b; Ho et al., 2020; Dhariwal and Nichol, 2021; Ho et al., 2022), or solving inverse problems (Kadkhodaie and Simoncelli, 2021; Cohen et al., 2021; Kawar et al., 2021; Laumont et al., 2022).

To model the conditional wavelet distribution $p(\bar{x}_j|x_j)$, we parameterize the score $\nabla_{\bar{y}_j} \log p(\bar{y}_j|x_j)$ of noisy wavelet coefficients $\bar{y}_j$ conditioned on a clean low-pass image x_j with a cCNN (eq. (4.3)) as in Chapter 3. Specifically, the cCNN takes as input three noisy wavelet detail channels, along with the corresponding low-pass channel, and generates three estimated detail channels. The network is trained to minimize mean square distance between $\bar{x}_j$ and $f_j(\bar{y}_j, x_j)$. Thanks to a conditional extension of eq. (4.5), an optimal network computes $f_j(\bar{y}_j, x_j) = \nabla_{\bar{y}_j} \log p(\bar{y}_j|x_j)$. Additionally, at the coarsest scale J, a CNN denoiser $f_J(y_J)$ is trained to estimate the score of the low-pass band, $\nabla_{y_J} \log p(x_J)$ by minimizing mean square distance between x_J and $f_J(y_J)$.

The following theorem proves that the Markov wavelet conditional property is equivalent to imposing that the cCNN RFs are restricted to the conditioning neighborhoods. The RF of a given element of the network response is defined as the set of input image pixels on which this element depends.

Theorem 4.1. *The wavelet conditional density $p(\bar{x}_j|x_j)$ is Markovian over a family of conditioning neighborhoods if and only if the conditional score $\nabla_{\bar{x}_j} \log p(\bar{x}_j|x_j)$ can be computed with a network whose RFs are included in these conditioning neighborhoods.*

The proof of the theorem is provided in Appendix C.1. Note that even if the conditional distribution of clean wavelet coefficients $p(\bar{x}|x_j)$ satisfies a local Markov property, the noisy distribution $p(\bar{y}_j|x_j)$ is in general not Markovian. However, we shall parameterize the scores with a cCNN with small RFs and hence show that both the noisy and clean distributions are Markovian. At each scale $1 \le j \le J$, the cCNN has RFs that are localized in both $\bar{y}_j$ and x_j, and have a fixed size over all scales, independent of the original image size. From Theorem 4.1, this defines the Markov conditioning neighborhoods of the learned model. The effect of the RF size is examined in the numerical experiments of Section 4.4.

Parameterizing the score with a convolutional network further implies that the conditional probability $p(\bar{x}_j|x_j)$ is stationary on the wavelet sampling lattice at scale j. Despite these strong

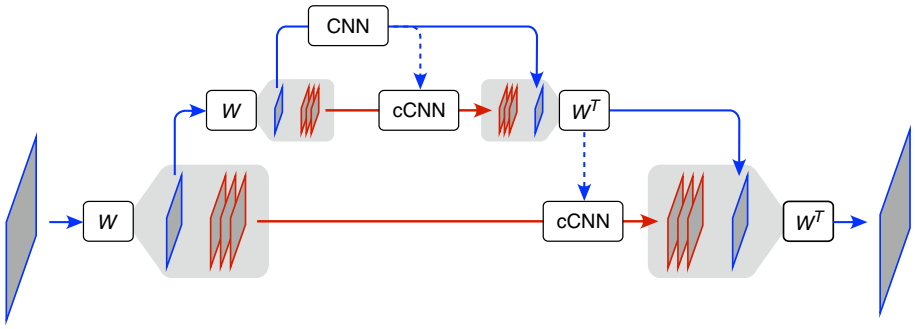

FIGURE 4.2: Wavelet conditional denoiser architecture used to estimate the score (illustrated for a two-scale decomposition). The input noisy image y (lower left) is decomposed by recursive application of a fast orthogonal wavelet transform W into successive low-pass images y_j (blue) and three wavelet detail images $\bar{y}_j$ (red). The coarsest low-pass image y_J is denoised using a CNN with a global receptive field to estimate $\hat{x}_J$. At all other scales, a local conditional CNN (cCNN) estimates $\hat{\bar{x}}_j$ from $\bar{y}_j$ conditioned on $\hat{x}_j$, from which W^T recovers $\hat{x}_{j-1}$.

simplifications, we shall see that these models models are able to capture complex long-range image dependencies in highly non-stationary image ensembles such as centered faces. This relies on the low-pass CNN, whose RF is designed to cover the entire image x_J, and thus does not enforce local Markov conditioning nor stationarity. The product density of eq. (4.1) is therefore not stationary.

4.4 Markov wavelet conditional denoising

We now evaluate our Markov wavelet conditional model on a denoising task. We use the trained CNNs to define a multiscale denoising architecture, illustrated in Figure 4.2. The wavelet transform of the input noisy image y is computed up to a coarse-scale J. The coarsest scale image is denoised by applying the denoising CNN learned previously: $\hat{x}_J = f_J(y_J)$. Then for $J \geq j \geq 1$, we compute the denoised wavelet coefficients conditioned on the previously estimated coarse image: $\hat{\bar{x}}_j = f(\bar{y}_j, \hat{x}_j)$. We then recover a denoised image at the next finer scale by applying an inverse wavelet transform: $\hat{x}_{j-1} = W^T(\hat{\bar{x}}_j, \hat{x}_j)$. At the finest scale we obtain the denoised image $\hat{x} = \hat{x}_0$.

Because of the orthogonality of W, the global MSE can be decomposed into a sum of wavelet MSEs at each scale, plus the coarsest scale error: $\|x - \hat{x}\|^2 = \sum_{j=1}^{J-1} \|\bar{x}_j - \hat{\bar{x}}_j\|^2 + \|x_J - \hat{x}_J\|^2$. The global MSE thus summarizes the precision of the score models computed over all scales. We evaluate the peak signal-to-noise ratio (PSNR) of the denoised image as a function of the noise level, expressed as the PSNR of the noisy image. We use the CelebA dataset (Liu et al., 2015) at 160×160 resolution. We use the simplest of all orthogonal wavelet decompositions, the Haar wavelet, constructed from separable filters that compute averages and differences of adjacent pairs of pixel values (Haar, 1910). All denoisers are "universal" (they can operate on images contaminated with noise of any standard deviation), and "blind" (they are not informed of the noise level). They all have the same depth and layer widths, and their receptive field size is controlled by changing the convolutional kernel size of each layer. Appendix C.3 provides architecture and training details.

Figure 4.3 shows that the multiscale denoiser based on a conditional wavelet Markov model outperforms a conventional denoiser that implements a Markov probability model in the pixel domain. More precisely, we observe that when the Markov structure is defined over image pixels, the performance degrades considerably with smaller RFs (Figure 4.3, left panel), especially at large noise levels (low PSNR). Images thus contain long-range global dependencies that cannot be captured by Markov models that are localized within the pixel lattice. On the other hand,

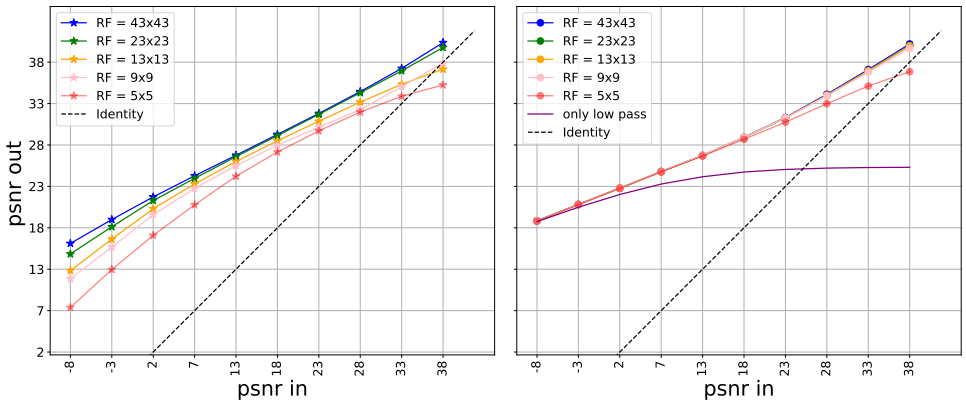

Figure 4.3: Comparison of denoiser performance on 160×160 *test images from the CelebA dataset. Each panel shows the error of the denoised image as a function of the noise level, both expressed as the peak signal-to-noise ratio (PSNR).* Left: *Conventional CNN denoisers with different RF sizes. The blue curve shows performance of BF-CNN (Mohan et al., 2019). The rest are BF-CNN variants with smaller RF obtained from setting some intermediate filter sizes to* 1×1. Right: *Multiscale denoisers, as depicted in Figure 4.2, with different cCNN RF sizes. Note that the low-pass denoiser RF is* 40×40 *in all cases, and thus covers the entire low-pass band.*

multiscale denoising performance remains nearly the same for RF sizes down to 9×9, and degrades for 5×5 RFs only at small noise levels (high PSNR) (Figure 4.3, right panel). This is remarkable considering that, for the finest scale, the 9×9 RF implies conditioning on one percent of the coefficients. The wavelet conditional score model thus successfully captures long-range image dependencies, even with small Markov neighborhoods.

It is also worth noting that in the large noise regime (i.e., low PSNR), all multiscale denoisers (even with RF as small as 5×5) significantly outperforms the conventional denoiser with the largest tested RF size (43×43). The dependency on RF size in this regime demonstrates the inadequacy of local modeling in the pixel domain. On the contrary, the effective neighborhoods of the multiscale denoiser are spatially global, but operate with spatially-varying resolution. Specifically, neighborhoods are of fixed size at each scale, but due to the subsampling, cover larger proportions of the image at coarser scales. The CNN applied to the coarsest low-pass band (scale J) is spatially global, and the denoising of this band alone explains the performance at the highest noise levels (magenta curve, Figure 4.3).

To further illustrate this point, consider the denoising examples shown in Figure 4.4. Since all denoisers are bias-free, they are piecewise linear (as opposed to piecewise affine), providing some interpretability (Mohan et al., 2019). Specifically, each denoised pixel is computed as an adaptively weighted sum over the noise input pixels. The last panels show the equivalent adaptive linear filter that was used to estimate the pixel marked by the green square, which can be estimated from the associated row of the Jacobian. The top row shows denoising results of a conventional CNN denoiser for small images that are the size of the network RF. Despite very heavy noise levels, the denoiser exploits the global structure of the image, and produces a result approximating the clean image. The second row shows the results after training the same denoiser architecture on much larger images. Now the adaptive filter is much smaller than the image, and the denoiser solution fails to capture the global structure of the face. Finally, the last row shows that the multiscale wavelet conditional denoiser can successfully approximate the global structure of a face despite the extreme levels of noise. Removing high levels of noise requires knowledge of global structure. In our multiscale conditional denoiser, this is achieved by setting the RF size of the low-pass denoiser equal to the entire low-pass image size, similarly to the denoiser shown on the top row. Then, each successive conditioning stage provides information at a finer resolution, over ever smaller RFs relative to the coefficient

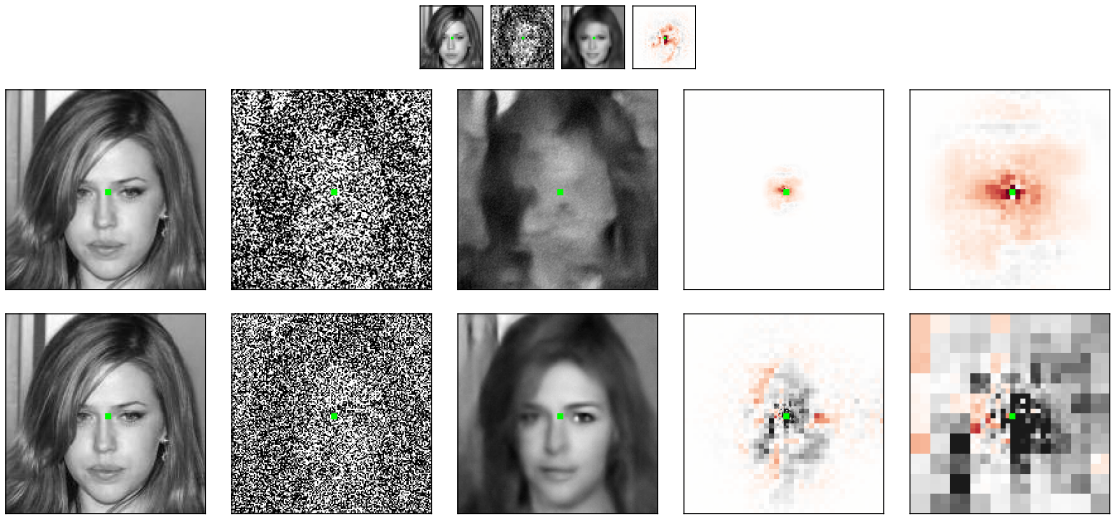

FIGURE 4.4: Denoising examples. Each row shows clean image, noisy image, denoised image, and the adaptive filter (one row of the Jacobian of the end-to-end denoising transformation) used by the denoiser to estimate a specific pixel, indicated in green. The heat-map ranges from red for most negative to black for most positive values. In the last two rows, the last column shows an enlargement of this adaptive filter, for enhanced visibility. Images are displayed proportional to their sizes. Top row: 40×40 *images estimated with a CNN denoiser with RF* 40×40. Second row: 160×160 *images estimated with a CNN denoiser with RF* 43×43. Third row: 160×160 *images, estimated with the proposed conditional multiscale denoiser of Figure 4.2. The denoiser uses a* 40×40 *RF for the coarsest scale, and* 13×13 *RFs for conditional denoising of subsequent finer scales.*

lattice. The adaptive filter shown in the last column has a *foveated* structure: the estimate of the marked pixel depends on all pixels in the noisy image, but those that are farther away are only included within averages over larger blocks. Thus, imposing locality in the wavelet domain lifts the curse of dimensionality without loss of performance, as opposed to a locality (Markov) assumption in the pixel domain.

4.5 Markov wavelet conditional super-resolution and synthesis

We generate samples from the learned wavelet conditional distributions in order to visually assess the quality of the model in a super-resolution task. We compare this approach with solving the super-resolution inverse problem directly using a CNN denoiser operating in the pixel domain. We also compare the models on image synthesis.

We first give a high-level description of our conditional generation algorithm. The low-resolution image x_J is used to conditionally generate wavelet coefficients $\bar{x}_J$ from the conditional distribution $p(\bar{x}_J|x_J)$. An inverse wavelet transform next recovers a higher-resolution image x_{J-1} from both x_J and $\bar{x}_J$. The conditional generation and wavelet reconstruction are repeated J times, increasing the resolution of the sample at each step. In the end, we obtain a sample x from the full-resolution image distribution conditioned on the starting low-resolution image $p(x|x_J)$. x is thus a stochastic super-resolution estimate of x_J.

To draw samples from the distributions $p(\bar{x}_j|x_j)$ implicitly embedded in the wavelet conditional denoisers, we use the algorithm of Kadkhodaie and Simoncelli (2021), which performs stochastic gradient ascent on the log-probability obtained from the cCNN using eq. (4.5). This is similar to score-based diffusion algorithms (Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021b), but the timestep hyper-parameters require essentially no tuning, since stepsizes are automatically obtained from the magnitude of the estimated score. Extension to the conditional case is straightforward. The sampling algorithm is detailed in Appendix C.4. All the cCCN

Figure 4.5: Super-resolution examples. Column 1*: original images (*320×320*).* Column 2*: Low-pass image of a 3-stage wavelet decomposition (downsampled to* 40×40*) expanded to full size for viewing.* Column 3*: Conditional generation of full-resolution images using CNN denoiser with RF of size* 43×43*.* Column 4*: Coarse-to-fine conditional generation using the multiscale cCNN denoisers, each with RFs of size* 13×13*.*

denoisers have a RF size of 13×13. Train and test images are from the CelebA HQ dataset (Karras et al., 2018) and of size 320×320. Samples drawn using the conditional denoiser correspond to a Markov conditional distribution with neighborhoods restricted to the RFs of the denoiser. We compare these with samples from a model with a local Markov neighbhorhood in the pixel domain. This is done using a CNN with a 40×40 RF trained to denoise full-resolution images, which approximates the score $\nabla \log p(x)$. Given the same low-pass image x_J, we can generate samples from $p(x|x_J)$ by viewing this as sampling from the image distribution of x constrained by a linear measurements x_J. This is done with the same sampling algorithm, with a small modification, again described in Appendix C.4.

Figure 4.5 shows super-resolution samples from these two learned image models. The local Markov model in the pixel domain generates details that are sharp but artifactual and incoherent over long spatial distances. On the other hand, the Markov wavelet conditional model produces much more natural-looking face images. This demonstrates the validity of our model: although these face images are not stationary (they have global structures shared across the dataset), and are not Markov in the pixel domain (there are clearly long-range dependencies that operate across the image), the *details* can be captured with local stationary Markov wavelet conditional distributions.

We also evaluated the Markov wavelet conditional model on image synthesis. We first synthesize a 40×40 terminal low-pass image using the score, $\nabla_{x_J} \log p(x_J)$, obtained from the low-pass CNN denoiser with a global RF. Again, unlike the conditional wavelet stages, this architectural choice does not enforce any local Markov structure nor stationarity. This global RF allows capturing global non-stationary structures, such as the overall face shape. The synthesis then proceeds using the same coarse-to-fine steps as used for super-resolution: wavelet coefficients at each successive scale are generated by drawing a sample using the cCNN conditioned

FIGURE 4.6: Image synthesis. Left four images: *Coarse-to-fine synthesis, achieved by sampling the score learned for each successive conditional distribution. Synthesized images are shown at four resolutions, from coarse-scale only (leftmost,* 40×40*) to the finest scale (rightmost,* 320×320*). Conditional RFs are all* 13×13*.* Right image: *Synthesis using a pixel-domain CNN with a receptive field (*40×40*) smaller than the synthesized image* 320×320*.*

on the previous scale.

The first (smallest) image in Figure 4.6 is generated from the low-pass CNN (see Appendix C.4 for algorithm). We can see that it successfully captures the coarse structure of a face. This image is then refined by application of successive stages of the multiscale super-resolution synthesis algorithm described above. The next three images in Figure 4.6 show successively higher resolution images generated in the procedure. For comparison, the last image in Figure 4.6 shows a sample generated using a conventional CNN with equivalent RFs trained on large face images. Once again, this illustrates that assuming spatially localized Markov property on the pixel lattice and estimating the score with a CNN with RF smaller than the image fails to capture the non-stationary distribution of faces. Specifically, the model is unable to generate structures larger than the RF size, and the samples are texture-like and composed of local regions resembling face parts.

We note that the quality of the generated images is not on par with the most recent score-based diffusion methods (e.g., Ramesh et al. (2022); Saharia et al. (2022); Rombach et al. (2022)), which have also been used for iterative super-resolution of an initial coarse-scale sample. These methods use much larger networks (more than a billion parameters, compared to ours which uses 600k parameters for the low-pass CNN and 200k for the cCNNs), and each scale-wise denoiser is itself a U-Net, with associated RF covering the entire image. Thus, the implicit probability models in these networks are global, and it is an open question whether and how these architectures are able to escape the curse of dimensionality. Our local conditional Markov assumptions provide a step towards the goal of making explicit the probability model and its factorization into low-dimensional components.

4.6 Discussion

In this chapter, we have generalized a Markov wavelet conditional probability model of image distributions, and developed an explicit implementation using cCNNs to estimate the conditional model scores. The resulting conditional wavelet distributions are stationary and Markov over neighborhoods corresponding to the cCNN receptive fields. The coarse-scale low-pass band is modeled using the score estimated with a CNN with global receptive fields. We trained this model on a dataset of face images, which are non-stationary with large-scale geometric features. We find that the model, even with relatively small cCNN RFs, succeeds in capturing these features, producing high-quality results on denoising and super-resolution tasks. We contrast this with local Markov models in the pixel domain, which are not able to capture these features, and are instead limited to stationary ergodic textures.

The Markov wavelet conditional model demonstrates that probability distributions of images

can be factorized as products of conditional distributions that are local. This model provides a mathematical justification which can partly explain the success of coarse-to-fine diffusion synthesis (Ho et al., 2020) which also computes conditional scores at each scale. Although we set out to understand how factorization of density models could allow them to avoid the curse of dimensionality in training, it is worth noting that the dimensionality of the conditioning neighborhoods in our network is still uncomfortably high ($4 \times 9 \times 9 = 324$). This is reduced by a factor of roughly 300 relative to the image size ($320 \times 320 = 102,400$), and this dimensionality remains constant even if this image size is increased, but it is still not sufficient to explain how the conditional score can be trained with realistic amounts of data. In addition, the terminal low-pass CNN operates globally (dimensionality $40 \times 40 = 1600$). Thus, the question of how to further reduce the overall dimensionality of the model remains open.

Our experiments were performed on cropped and centered face images, which present a particular challenge given their obvious non-stationarity. The conditional models are approximately stationary due to the fully convolutional structure of the cCNN operations (although this is partially violated by zero-padded boundary handling). As such, the non-stationary capabilities of the full model arise primarily from the terminal low-pass CNN, which uses spatially global RFs. We speculate that for more diverse image datasets (e.g., a large set of natural images), a much larger capacity low-pass CNN will be needed to capture global structures. This is consistent with current deep networks that generate high-quality synthetic images using extremely large networks (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022). On the other hand, the cCNNs in our model all share the same architecture and local RFs, and may (empirically) be capturing similar local conditional structure at each scale. Forcing these conditional densities to be the same at each scale (through weight sharing of the corresponding cCNNs) would impose a scale invariance assumption on the overall model. This would further reduce model complexity, and enable synthesis and inference on images of size well beyond that of the training set.

In this chapter, together with Chapters 2 and 3, we have shown that a wavelet conditional factorization may reveal properties of log-concavity, regularity, and locality that were not enjoyed by the global distribution $p(x)$. These results evidence some multiscale properties of image distributions and how they can partially be leveraged to alleviate the curse of dimensionality. The central issue remains to explain how score-based diffusion models manage to estimate the scores of these distributions and seemingly generalize from small amounts of data. These issues are the same as in supervised learning, though from a different point of view, and thus one might expect that studying the score estimation problem might bring new insights into the classical problem of generalization in supervised learning.

Part II

Non-Linear Operators for Image Classification

Chapter

5 Separation and Concentration in Deep Networks

Chapter content

5.1 Introduction 72

5.2 Classification by separation and concentration 72

5.2.1 Tight frame rectification and thresholding 72

5.2.2 Two-layer networks without bias 75

5.3 Deep learning by scattering and concentrating 77

5.3.1 Scattering cascade of wavelet frame separations 77

5.3.2 Separation and concentration in learned scattering networks 79

5.4 Discussion 81

Numerical experiments demonstrate that deep neural network classifiers progressively separate class distributions around their mean, achieving linear separability on the training set, and increasing the Fisher discriminant ratio. We explain this mechanism with two types of non-linear operators. We prove that a rectifier without biases applied to sign-invariant tight frames can separate class means and increase Fisher ratios. On the opposite, a soft-thresholding on tight frames can reduce within-class variabilities while preserving class means. Variance reduction bounds are proved for Gaussian mixture models. We show that separation of class means can be achieved with rectified wavelet tight frames that are not learned. It defines a scattering transform. Learning 1×1 convolutional tight frames along scattering channels and applying a soft-thresholding reduces within-class variabilities. The resulting scattering network reaches the classification accuracy of ResNet-18 on CIFAR-10 and ImageNet, with fewer layers and no learned biases.

This chapter is adapted from the following publication: John Zarka, Florentin Guth, and Stéphane Mallat. Separation and concentration in deep networks. In *International Conference on Learning Representations*, 2021. We omit the proof of Theorem 5.1, which was not done by the author of this dissertation. This chapter was written before Chapter 6 and is therefore less mature. In particular, the role of phase collapse in separating class means was not yet fully identified. Separation and concentration operators are treated symmetrically in this chapter, whereas the results in Chapter 6 show that separation with phase collapses alone is sufficient to achieve high classification accuracies, which is not the case for concentration operators. The learned scattering architectures introduced in this chapter are thus precursors for the more streamlined one of Chapter 6. Finally, we note that the theoretical and numerical results presented in this chapter have been refined by Zarka (2022), who notably presents generalizations of Theorem 5.1.

5.1 Introduction

Several numerical works (Oyallon, 2017; Papyan, 2020; Papyan et al., 2020) have shown that deep neural networks classifiers (LeCun et al., 2015) progressively concentrate each class around separated means, until the last layer, where within-classes variability may nearly "collapse" (Papyan et al., 2020). The linear separability of a class mixture is characterized by the Fisher discriminant ratio (Fisher, 1936; Rao, 1948). The Fisher discriminant ratio measures the separation of class means relatively to the variability within each class, as measured by their covariances. The neural collapse appears through a considerable increase of the Fisher discriminant ratio during training (Papyan et al., 2020). No mathematical mechanism has yet been provided to explain this separation and concentration of probability measures.

Linear separability and Fisher ratios can be increased by separating class means without increasing the variability of each class, or by concentrating each class around its mean while preserving the mean separation. This chapter shows that these separation or concentration properties can be achieved with one-layer network operators using different pointwise non-linearities. We cascade these operators to define structured deep neural networks with high classification accuracies, and which can be analyzed mathematically.

Section 5.2 studies two-layer networks computed with a linear classifier applied to ρD^{T}, where D^{T} is linear and ρ is a pointwise non-linearity. First, we show that ρD^{T} can separate class means with a ReLU $\rho_r(u) = \max(u, 0)$ and a sign-invariant D. We prove that $\rho_r D^{\mathrm{T}}$ then increases the Fisher ratio. As in Parseval networks (Cisse et al., 2017), D is normalized by imposing that it is a tight frame which satisfies $DD^{\mathrm{T}} = \mathrm{Id}$. Second, to concentrate the variability of each class around its mean, we use a shrinking non-linearity implemented by a soft-thresholding ρ_t. For Gaussian mixture models, we prove that $\rho_t D^{\mathrm{T}}$ concentrates within-class variabilities while nearly preserving class means, under appropriate sparsity hypotheses. A linear classifier applied to these ρD^{T} defines two-layer neural networks with no learned bias parameters in the hidden layer, whose properties are studied mathematically and numerically.

Cascading several convolutional tight frames with ReLUs or soft-thresholdings defines a deep neural network which progressively separates class means and concentrates their variability. One may wonder if we can avoid learning these frames by using prior information on the geometry of images. Section 5.3 shows that the class mean separation can be computed with wavelet tight frames, which are not learned. They separate scales, directions and phases, which are known groups of transformations. A cascade of wavelet filters and rectifiers defines a scattering transform (Mallat, 2012), which has previously been applied to image classification (Bruna and Mallat, 2013; Oyallon and Mallat, 2015). However, such networks do not reach state-of-the-art classification results. We show that important improvements are obtained by learning 1×1 convolutional projectors and tight frames, which concentrate within-class variabilities with soft-thresholdings. It defines a bias-free deep scattering network whose classification accuracy reaches ResNet-18 (He et al., 2016) on CIFAR-10 and ImageNet.

5.2 Classification by separation and concentration

The last hidden layer of a neural network defines a representation $\Phi(x)$, to which is applied a linear classifier. This section studies the separation of class means and class variability concentration for $\Phi = \rho D^{\mathrm{T}}$ in a two-layer network.

5.2.1 Tight frame rectification and thresholding

We begin by briefly reviewing the properties of linear classifiers and Fisher discriminant ratios. We then analyze the separation and concentration of $\Phi = \rho D^{\mathrm{T}}$, when ρ is a rectifier or a soft-thresholding and D is a tight frame.

Linear classification and Fisher ratio. We consider a random data vector $x \in \mathbb{R}^d$ whose class labels are $y(x) \in \{1, ..., C\}$. Let x_c be a random vector representing the class c, whose probability distribution is the distribution of x conditioned by $y(x) = c$. We suppose that all classes are equiprobable for simplicity. Ave_c denotes $C^{-1} \sum_{c=1}^{C}$.

We compute a representation of x with an operator Φ which is standardized, so that $\mathbb{E}(\Phi(x)) = 0$ and each coefficient of $\Phi(x)$ has a unit variance. The class means $\mu_c = \mathbb{E}[\Phi(x_c)]$ thus satisfy $\sum_c \mu_c = 0$. A linear classifier (θ, b) on $\Phi(x)$ returns the index of the maximum coordinate of $\theta^{\mathrm{T}}\Phi(x) + b \in \mathbb{R}^C$. An optimal linear classifier (θ, b) minimizes the probability of a classification error. Optimal linear classifiers are estimated by minimizing a regularized loss function on the training data. Neural networks often use logistic linear classifiers, which minimize a cross-entropy loss. The standardization of the last layer $\Phi(x)$ is implemented with a batch normalization (Ioffe and Szegedy, 2015).

A linear classifier can have a small error if the typical sets of each $\Phi(x_c)$ have little overlap, and in particular if the class means $\mu_c = \mathbb{E}[\Phi(x_c)]$ are sufficiently separated relatively to the variability of each class. Under the Gaussian hypothesis, the variability of each class is measured by the covariance Σ_c of $\Phi(x_c)$. Let $\Sigma_W = \text{Ave}_c \, \Sigma_c$ be the average within-class covariance and $\Sigma_B = \text{Ave}_c \, \mu_c \, \mu_c^{\mathrm{T}}$ be the between-class covariance of the means. The within-class covariance can be whitened and normalized to Id by transforming $\Phi(x)$ with the square root $\Sigma_W^{-1/2}$ of Σ_W^{-1}. All classes c, c' are highly separated if $\|\Sigma_W^{-1/2}\mu_c - \Sigma_W^{-1/2}\mu_{c'}\| \gg 1$. This separation is captured by the Fisher discriminant ratio $\Sigma_W^{-1}\Sigma_B$. We shall measure its trace

$$C^{-1} \operatorname{tr}(\Sigma_W^{-1}\Sigma_B) = \operatorname*{Ave}_c \|\Sigma_W^{-1/2}\mu_c\|^2. \tag{5.1}$$

Fisher ratios have been used to train deep neural networks as a replacement for the cross-entropy loss (Dorfer et al., 2015; Stuhlsatz et al., 2012; Sun et al., 2019; Wu et al., 2017; Sultana et al., 2018; Li et al., 2016). In this chapter, we use their analytic expression to analyze the improvement of linear classifiers.

Linear classification obviously cannot be improved with a linear representation Φ. The following proposition gives a simple condition to improve (or maintain) the error of linear classifiers with a non-linear representation.

Proposition 5.1. *If Φ has a linear inverse, then it decreases (or maintains) the error of the optimal linear classifier, and it increases (or maintains) the Fisher ratio (5.1).*

To prove this result, observe that if Φ has a linear inverse Φ^{-1} then $\theta^{\mathrm{T}}x = \theta'^{\mathrm{T}}\Phi(x)$ with $\theta'^{\mathrm{T}} = \theta^{\mathrm{T}}\Phi^{-1}$. The minimum classification error by optimizing θ is thus above the error obtained by optimizing θ'. Appendix D.1 proves that the Fisher ratio (5.1) is also increased or preserved.

There are qualitatively two types of non-linear operators that increase the Fisher ratio $\Sigma_W^{-1}\Sigma_B$. Separation operators typically increase the distance between the class means without increasing the variance Σ_W within each class. We first study such operators having a linear inverse, which guarantees through Proposition 5.1 that they increase the Fisher ratio. We then study concentration operators which reduce the variability Σ_W with non-linear shrinking operators, which are not invertible. It will thus require a finer analysis of their properties.

Separation by tight frame rectification. Let $\Phi = \rho D^{\mathrm{T}}$ be an operator which computes the first layer of a neural network, where ρ is a pointwise non-linearity and D is linear. We first study separation operators computed with a ReLU $\rho_r(u) = \max(u, 0)$ applied to an invertible sign-invariant matrix D. Such a matrix has columns that can be regrouped in pairs of opposite signs. It can thus be written $D = [-\tilde{D}, \tilde{D}]$ where $\tilde{D}$ is invertible. The operator ρD^{T} separates coefficients according to their sign. Since $\rho_r(u) - \rho_r(-u) = u$, it results that $\Phi = \rho_r D^{\mathrm{T}}$ is linearly invertible. According to Proposition 5.1, it increases (or maintains) the Fisher ratio, and we want to choose D to maximize this increase.

Observe that $\rho_r(\alpha u) = \alpha \rho_r(u)$ if $\alpha \geq 0$. We can thus normalize the columns d_m of D without affecting linear classification performance. To ensure that $D \in \mathbb{R}^{d \times p}$ is invertible with a stable inverse, we impose that it is a normalized tight frame of $\mathbb{R}^d$ satisfying

$$DD^{\mathrm{T}} = \mathrm{Id} \quad \text{and} \quad \|d_m\|^2 = d/p \text{ for } 1 \leq m \leq p.$$

The tight frame can be interpreted as a rotation operator in a higher dimensional space, which aligns the axes and the directions along which ρ_r performs the sign separation. This rotation must be adapted in order to optimize the separation of class means. The fact that D is a tight frame can be interpreted as a normalization which simplifies the mathematical analysis.

Suppose that all classes x_c of x have a Gaussian distribution with a zero mean $\mu_c = 0$, but different covariances Σ_c. These classes are not linearly separable because they have the same mean, and the Fisher ratio is 0. Applying $\rho_r D^{\mathrm{T}}$ can separate these classes and improve the Fisher ratio. Indeed, if z is a zero-mean Gaussian random variable, then $\mathbb{E}[\max(z, 0)] = (2\pi)^{-1/2}\mathbb{E}[z^2]^{1/2}$ so we verify that for $D = [-\tilde{D}, \tilde{D}]$,

$$\mathbb{E}\Big[\rho_r D^{\mathrm{T}} x_c\Big] = (2\pi)^{-1/2}\Big(\mathrm{diag}(\tilde{D}^{\mathrm{T}}\Sigma_c\tilde{D})^{1/2}, \mathrm{diag}(\tilde{D}^{\mathrm{T}}\Sigma_c\tilde{D})^{1/2}\Big).$$

The Fisher ratio can then be optimized by maximizing the covariance Σ_B between the mean vector components $\mathrm{diag}(\tilde{D}^{\mathrm{T}}\Sigma_c D)^{1/2}$ for all classes c. If we know a priori that that x_c and $-x_c$ have the same probability distribution, as in the Gaussian example, then we can replace ρ_r by the absolute value $\rho_a(u) = |u| = \rho_r(u) + \rho_r(-u)$, and $\rho_r D^{\mathrm{T}}$ by $\rho_a \tilde{D}^{\mathrm{T}}$, which reduces by 2 the frame size.

Concentration by tight frame soft-thresholding. If the class means of x are already separated, then we can increase the Fisher ratio with a non-linear Φ that concentrates each class around its mean. The operator Φ must reduce the within-class variance while preserving the class separation. This can be interpreted as a non-linear noise removal if we consider the within-class variability as an additive noise relatively to the class mean. It can be done with soft-thresholding estimators introduced in Donoho and Johnstone (1994). A soft-thresholding $\rho_t(u) = \mathrm{sign}(u)\,\max(|u| - \lambda, 0)$ shrinks the amplitude of u by λ in order to reduce its variance, while introducing a bias that depends on λ. Donoho and Johnstone (1994) proved that soft-thresholding estimators are highly effective to estimate signals that have a sparse representation in a tight frame D, which then plays the role of a dictionary.

To evaluate more easily the effect of a tight frame soft-thresholding on the class means, we apply the linear reconstruction D on $\rho_t D^{\mathrm{T}} x$, which thus defines a representation $\Phi(x) = D\rho_t D^{\mathrm{T}} x$. For a strictly positive threshold, this operator is not invertible, so we cannot apply Proposition 5.1 to prove that the Fisher ratio increases. We study directly the impact of Φ on the mean and covariance of each class. Let x_c be the vector representing the class c. The mean $\mu_c = \mathbb{E}[x_c]$ is transformed into $\bar{\mu}_c = \mathbb{E}[\Phi(x_c)]$ and the covariance Σ_c of x_c into the covariance $\overline{\Sigma}_c$ of $\Phi(x_c)$. The average covariances are $\Sigma_W = \mathrm{Ave}_c\,\Sigma_c$ and $\overline{\Sigma}_W = \mathrm{Ave}_c\,\overline{\Sigma}_c$.

Suppose that each x_c is a Gaussian mixture, with a potentially large number of Gaussian components centered at $\mu_{c,k}$ with a fixed covariance $\sigma^2\,\mathrm{Id}$:

$$p_c = \sum_k \pi_{c,k}\,\mathcal{N}(\mu_{c,k}, \sigma^2\,\mathrm{Id}). \tag{5.2}$$

This model is quite general, since it amounts to covering the typical set of realizations of x_c with a union of balls of radius σ, centered in the $(\mu_{c,k})_k$. The following theorem relates the reduction of within-class covariance to the sparsity of $D^{\mathrm{T}}\mu_{c,k}$. It relies on the soft-thresholding estimation results of Donoho and Johnstone (1994).

For simplicity, we suppose that the tight frame is an orthogonal basis, but the result can be extended to general normalized tight frames. The sparsity is expressed through the decay of sorted basis coefficients. For a vector $z \in \mathbb{R}^d$, we denote $z^{(r)}$ a coefficient of rank r: $|z^{(r)}| \geq |z^{(r+1)}|$ for $1 \leq r \leq d$. The theorem imposes a condition on the amplitude decay of the $(D^{\mathrm{T}}\mu_{c,k})^{(r)}$ when r increases, which is a sparsity measure. We write $a(r) \sim b(r)$ if $C_1\, a(r) \leq b(r) \leq C_2\, a(r)$ where C_1 and C_2 do not depend upon d nor σ. The theorem derives upper bounds on the reduction of within-class covariances and on the displacements of class means. The constants do not depend upon d when it increases to ∞ nor on σ when it decreases to 0.

Theorem 5.1. *Under the mixture model hypothesis (5.2), we have*

$$\mathrm{tr}(\Sigma_W) = \mathrm{tr}(\Sigma_M) + \sigma^2\, d, \quad \textit{with} \quad \mathrm{tr}(\Sigma_M) = C^{-1} \sum_{c,k} \pi_{c,k}\, \|\mu_c - \mu_{c,k}\|^2. \tag{5.3}$$

If there exists $s > 1/2$ such that $|(D^{\mathrm{T}}\mu_{c,k})^{(r)}| \sim r^{-s}$ then a tight frame soft-thresholding with threshold $\lambda = \sigma\sqrt{2\log d}$ satisfies

$$\mathrm{tr}(\overline{\Sigma}_W) = 2\,\mathrm{tr}(\Sigma_M) + O(\sigma^{2-1/s}\, \log d), \tag{5.4}$$

and all class means satisfy

$$\|\mu_c - \overline{\mu}_c\|^2 = O(\sigma^{2-1/s}\, \log d). \tag{5.5}$$

The proof is in the original publication (Zarka et al., 2021, Appendix B). Under appropriate sparsity hypotheses, the theorem proves that applying $\Phi = D\rho_t D^{\mathrm{T}}$ reduces considerably the trace of the within-class covariance. The Gaussian variance $\sigma^2 d$ is dominant in (5.3) and is reduced to $O(\sigma^{2-1/s}\, \log d)$ in (5.4). The upper bound (5.5) also proves that $D\rho_t D^{\mathrm{T}}$ creates a relatively small displacement of class means, which is proportional to $\log d$. This ensures that all class means remain well separated. These bounds qualitatively explains the increase of Fisher ratios, but they are not sufficient to prove a precise bound on these ratios.

In numerical experiments, the threshold value of the theorem is automatically adjusted as follows. Non-asymptotic optimal threshold values have been tabulated as a function of d by Donoho and Johnstone (1994). For the range of d used in our applications, a nearly optimal threshold is $\lambda = 1.5\,\sigma$. We rescale the frame variance σ^2 by standardizing the input x so that it has a zero mean and each coefficient has a unit variance. In high dimension d, the within-class variance typically dominates the variance between class means. Under the unit variance assumption we have $\mathrm{tr}(\Sigma_W) \approx d$. If $D \in \mathbb{R}^{d\times p}$ is a normalized tight frame then we also verify as in (5.3) that $\mathrm{tr}(\Sigma_W) \approx \sigma^2 p$ so $\sigma^2 \approx d/p$. It results that we choose $\lambda = 1.5\,\sqrt{d/p}$.

A soft-thresholding can also be computed from a ReLU with threshold $\rho_{rt}(u) = \max(u-\lambda, 0)$ because $\rho_t(u) = \rho_{rt}(u) - \rho_{rt}(-u)$. It results that $[D, -D]\,\rho_{rt}\,[D, -D]^{\mathrm{T}} = D\,\rho_t\,D^{\mathrm{T}}$. However, a thresholded rectifier has more flexibility than a soft-thresholding, because it may recombine differently $\rho_{rt}D^{\mathrm{T}}$ and $\rho_{rt}(-D^{\mathrm{T}})$ to also separate class means, as explained previously. The choice of threshold then becomes a trade-off between separation of class means and concentration of class variability. In numerical experiments, we choose a lower $\lambda = \sqrt{d/p}$ for a ReLU with a threshold.

5.2.2 Two-layer networks without bias

We study two-layer bias-free networks that implement a linear classification on ρD^{T}, where D is a normalized tight frame and ρ may be a rectifier, an absolute value or a soft-thresholding, with no learned bias parameter. Bias-free networks have been introduced for denoising in Mohan et al. (2019), as opposed to classification or regression. We show that such bias-free networks have a limited expressivity and do *not* satisfy universal approximation theorems (Pinkus, 1999; Bach,

2017a). However, numerical results indicate that their separation and contractions capabilities are sufficient to reach similar classification results as two-layer networks with biases on standard image datasets.

Applying a linear classifier on $\Phi(x)$ computes

$$\theta^{\mathrm{T}}\Phi(x) + b = \theta^{\mathrm{T}}\rho D^{\mathrm{T}}x + b.$$

This two-layer neural network has no learned bias parameters in the hidden layer, and we impose that $DD^{\mathrm{T}} = \mathrm{Id}$ with atoms (the columns of D) $(d_m)_m$ having constant norms. As a result, the following theorem proves that it does not satisfy the universal approximation theorem. We define a binary classification problem for which the probability of error remains above $1/4$ for any number p of neurons in the hidden layer. The proof is provided in Appendix D.2 for a ReLU ρ_{rt} with any threshold. The theorem remains valid with an absolute value ρ_a or a soft-thresholding ρ_t, because they are linear combinations of ρ_{rt}.

Theorem 5.2. *Let $\lambda \geq 0$ be a fixed threshold and $\rho_{rt}(u) = \max(u - \lambda, 0)$. Let $\mathcal{D}$ be the set of matrices $D \in \mathbb{R}^{d\times p}$ with bounded columns $\|d_m\| \leq 1$. There exists a random vector $x \in \mathbb{R}^d$ which admits a probability density supported on the unit ball, and a C^∞ function $h\colon \mathbb{R}^d \to \mathbb{R}$ such that, for all $p \geq d$,*

$$\inf_{\theta\in\mathbb{R}^p, D\in\mathcal{D}, b\in\mathbb{R}} \mathbb{P}\Big[\operatorname{sgn}\Big(\theta^{\mathrm{T}}\rho_{rt}D^{\mathrm{T}}x + b\Big) \neq \operatorname{sgn}(h(x))\Big] \geq \frac{1}{4}.$$

Optimization. The parameters θ, D and b are optimized with a stochastic gradient descent that minimizes a logistic cross-entropy loss on the output. To impose $DD^{\mathrm{T}} = \mathrm{Id}$, following the optimization of Parseval networks (Cisse et al., 2017), after each gradient update of all network parameters, we insert a second gradient step to minimize $\alpha/2\,\|DD^{\mathrm{T}} - \mathrm{Id}\,\|^2$. This gradient update is

$$D \leftarrow (1+\alpha)D - \alpha DD^{\mathrm{T}}D. \tag{5.6}$$

We also make sure after every Parseval step that each atom d_m keeps a constant norm $\|d_m\| = \sqrt{d/p}$ by applying a spherical projection: $d_m \leftarrow \sqrt{d/p}\, d_m/\|d_m\|$. These steps are performed across all experiments described in the chapter, which ensures that all singular values of every learned tight frame are comprised between 0.99 and 1.01.

To reduce the number of parameters of the classification matrix $\theta^{\mathrm{T}} \in \mathbb{R}^{C\times p}$, we factorize $\theta^{\mathrm{T}} = \theta'^{\mathrm{T}} D$ with $\theta'^{\mathrm{T}} \in \mathbb{R}^{C\times d}$. It amounts to reprojecting ρD^{T} in $\mathbb{R}^d$ with the semi-orthogonal frame synthesis D, and thus defines

$$\Phi(x) = D\,\rho\,D^{\mathrm{T}}x.$$

A batch normalization is introduced after Φ to stabilize the learning of θ'.

Image classification by separation and concentration. Image classification is first evaluated on the MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky, 2009) image datasets. Table 5.1 gives the results of logistic classifiers applied to the input signal x and to $\Phi(x) = D\rho D^{\mathrm{T}}x$ for 3 different non-linearities ρ: absolute value ρ_a, soft-thresholding ρ_t, and ReLU with threshold ρ_{rt}. The tight frame D^{T} is a convolution on patches of size $k \times k$ with a stride of $k/2$, with $k = 14$ for MNIST and $k = 8$ for CIFAR. The tight frame D^{T} maps each patch to a vector of larger dimension, specified in Appendix D.3.

On each dataset, applying $D\rho D^{\mathrm{T}}$ on x greatly reduces linear classification error, which also appears with an increase of the Fisher ratio. For MNIST, all non-linearities produce nearly the same classification accuracy, but on CIFAR, the soft-thresholding has a higher error. Indeed, the

	$\Phi(x)$	x	$D\rho D^{\mathrm{T}}x$			$S_T(x)$
			$\rho = \rho_a$	$\rho = \rho_t$	$\rho = \rho_{rt}$	
MNIST	Error (%)	7.4	1.3	1.4	1.3	0.8
	Fisher	19	68	69	67	130
CIFAR	Error (%)	60.5	28.1	34.8	26.5	27.7
	Fisher	6.7	15	13	16	12

Table 5.1: For MNIST and CIFAR-10, the first row gives the logistic classification error and the second row the Fisher ratio (5.1), for different signal representations $\Phi(x)$. Results are evaluated with an absolute value ρ_a, a soft-thresholding ρ_t, and a ReLU with threshold ρ_{rt}.

class means of MNIST are distinct averaged digits, which are well separated, because all digits are centered in the image. Concentrating variability with a soft-thresholding is then sufficient. On the opposite, the classes of CIFAR images define nearly stationary random vectors because of arbitrary translations. As a consequence, the class means μ_c are nearly constant images, which are only discriminated by their average color. Separating these class means is then important for improving classification. As explained in Section 5.2.1, this is done by a ReLU ρ_r, or in this case an absolute value ρ_a, which reduces the error. The ReLU with threshold ρ_{rt} can interpolate between mean separation and variability concentration, and thus performs usually at least as well as the other non-linearities.

The error of the bias-free networks with a ReLU and an absolute value are similar to the errors obtained by training two-layer networks of similar sizes but with bias parameters: 1.6% error on MNIST (Simard et al., 2003), and 25% on CIFAR-10 (Krizhevsky, 2010). It indicates that the elimination of bias parameters does not affect performances, despite the existence of the counter-examples from Theorem 5.2 that cannot be well approximated by such architectures. This means that image classification problems have more structure that are not captured by these counter-examples, and that completeness in linear high-dimensional functional spaces may not be key mathematical properties to explain the preformances of neural networks. Figure 5.1 shows that the learned convolutional tight frames include oriented oscillatory filters, which is also often the case of the first layer of deeper networks (Krizhevsky et al., 2012). They resemble wavelet frames, which are studied in the next section.

5.3 Deep learning by scattering and concentrating

To improve classification accuracy, we cascade mean separation and variability concentration operators, implemented by ReLUs and soft-thresholdings on tight frames. This defines deep convolutional networks. However, we show that some tight frames do not need to be learned. Section 5.3.1 reviews scattering trees, which perform mean separation by cascading ReLUs on wavelet tight frames. Section 5.3.2 shows that we reach high classification accuracies by learning projectors and tight frame soft-thresholdings, which concentrate within-class variabilities along scattering channels.

5.3.1 Scattering cascade of wavelet frame separations

Scattering transforms have been introduced to classify images by cascading predefined wavelet filters with a modulus or a rectifier non-linearity (Bruna and Mallat, 2013). We write it as a product of wavelet tight frame rectifications, which progressively separate class means.

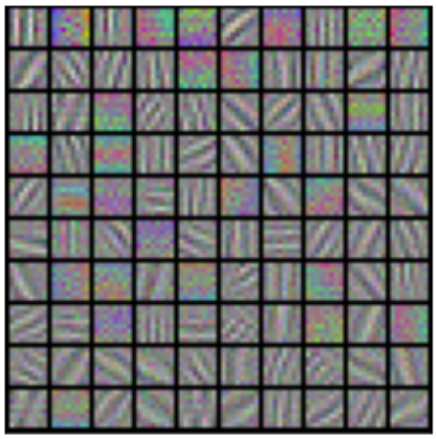

FIGURE 5.1: Examples of filters d_m from the convolutional tight frame D learned directly on the input x for CIFAR-10, using an absolute value non-linearity ρ_a. They resemble wavelet filters.

Wavelet frame. A wavelet frame separates image variations at different scales, directions and phases, with a cascade of filterings and subsamplings. We use steerable wavelets (Simoncelli and Freeman, 1995) computed with Morlet filters (Bruna and Mallat, 2013). There is one low-pass filter g_0, and L complex band-pass filters g_ℓ having an angular direction $\ell\pi/L$ for $0 < \ell \leq L$. These filters can be adjusted (Selesnick et al., 2005) so that the filtering and subsampling

$$\tilde{W}x(n,\ell) = x \star g_\ell(2n)$$

defines a complex tight frame $\tilde{W}$. Fast multiscale wavelet transforms are computed by cascading the filter bank $\tilde{W}$ on the output of the low-pass filter g_0 (Mallat, 2008).

Each complex filter g_ℓ is analytic, and thus has a real part and imaginary part whose phases are shifted by $\alpha = \pi/2$. This property is important to preserve equivariance to translation despite the subsampling with a stride of 2 (Selesnick et al., 2005). To define a sign-invariant frame as in Section 5.2.1, we must incorporate filters of opposite signs, which amounts to shifting their phase by π. We thus associate to $\tilde{W}$ a real sign-invariant tight frame W by considering separately the four phases $\alpha = 0, \pi/2, \pi, 3\pi/2$. It is defined by

$$Wx(n,\ell,\alpha) = x \star g_{\ell,\alpha}(2n),$$

with $g_{\ell,0} = 2^{-1/2}\text{Real}(g_\ell)$, $g_{\ell,\pi/2} = 2^{-1/2}\text{Imag}(g_\ell)$ and $g_{\ell,\alpha+\pi} = -g_\ell$. We apply a rectifier ρ_r to the output of all real band-pass filters $g_{\ell,\alpha}$ but not to the low-pass filter:

$$\rho_r W = \Big(x \star g_0(2n)\ ,\ \rho_r(x \star g_{\ell,\alpha}(2n))\Big)_{n,\alpha,\ell}.$$

The use of wavelet phase parameters with rectifiers is studied in Mallat et al. (2019). The operator $\rho_r W$ is linearly invertible because W is a tight frame and the ReLU is applied to band-pass filters, which come in pairs of opposite sign. Since there are 4 phases and a subsampling with a stride of 2, Wx is $(L + 1/4)$ times larger than x.

Scattering tree. A full scattering tree S_T of depth J is computed by iterating J times over $\rho_r W$. Since each $\rho_r W$ has a linear inverse, Proposition 5.1 proves that this separation can only increase the Fisher ratio. However it also increases the signal size by $(L + 1/4)^J$, which is

	Φ		S_T	S_P	$S_C\ (\rho_t)$	$S_C\ (\rho_{rt})$	ResNet
CIFAR	Error (%)		27.7	12.8	8.0	7.6	8.8
	Fisher		12	20	43	41	-
ImageNet	Error (%)	Top-5	54.1	20.5	11.6	10.7	10.9
		Top-1	73.0	42.3	31.4	29.7	30.2
	Fisher		2.0	18	51	44	-

TABLE 5.2: Linear classification error and Fisher ratios (5.1) of several scattering representations, on CIFAR-10 and ImageNet. For S_C, results are evaluated with a soft-thresholding ρ_t and a thresholded rectifier ρ_{rt}. The last column gives the error of ResNet-20 for CIFAR-10 (He et al., 2016) and ResNet-18 for ImageNet, taken from https://pytorch.org/docs/stable/torchvision/models.html.

typically much too large. This is avoided with orthogonal projectors, which perform a dimension reduction after applying each $\rho_r W$.

A pruned scattering tree S_T of depth J and order o is defined in Bruna and Mallat (2013) as a convolutional tree which cascades J rectified wavelet filter banks, and at each depth prunes the branches with P_j to prevent an exponential growth:

$$S_T = \prod_{j=1}^{J} P_j \, \rho_r \, W. \tag{5.7}$$

After the ReLU, the pruning operator P_j eliminates the branches of the scattering which cascade more than o band-pass filters and rectifiers, where o is the scattering order (Bruna and Mallat, 2013). After J cascades, the remaining channels have thus been filtered by at least $J - o$ successive low-pass filters g_0. We shall use a scattering transform of order $o = 2$. The operator P_j also averages the rectified output of the filters $g_{\ell,\alpha}$ along the phase α, for ℓ fixed. This averaging eliminates the phase. It approximatively computes a complex modulus and produces a localized translation invariance. The resulting pruning and phase average operator P_j is a 1×1 convolutional operator, which reduces the dimension of scattering channels with an orthogonal projection. If x has d pixels, then $S_T(x)[n, k]$ is an array of images having $2^{-2J}d$ pixels at each channel k, because of the J subsamplings with a stride of 2. The total number of channels K is $1 + JL + J(J-1)L^2/2$. Numerical experiments are performed with wavelet filters which approximate Gabor wavelets (Bruna and Mallat, 2013), with $L = 8$ directions. The number of scales J depends upon the image size. It is $J = 3$ for MNIST and CIFAR, and $J = 4$ for ImageNet, resulting in respectively $K = 217$, 651 and 1251 channels.

Each $\rho_r W$ can only improve the Fisher ratio and the linear classification accuracy, but it is not guaranteed that this remains valid after applying P_j. Table 5.1 gives the classification error of a logistic classifier applied on $S_T(x)$, after a 1×1 orthogonal projection to reduce the number of channels, and a spatial normalization. This error is almost twice smaller than a two-layer neural network on MNIST, given in Table 5.1, but it does not improve the error on CIFAR. On CIFAR, the error obtained by a ResNet-20 is 3 times lower than the one of a classifier on $S_T(x)$. The main issue is now to understand where this inefficiency comes from.

5.3.2 Separation and concentration in learned scattering networks

A scattering tree iteratively separates class means with wavelet filters. Its dimension is reduced by predefined projection operators, which may decrease the Fisher ratio and linear separability. To avoid this source of inefficiency, we define a scattering network which learns these projections. The second step introduces tight frame thresholdings along scattering channels, to concentrate within-class variabilities. Image classification results are evaluated on the CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) datasets.

CIFAR	Layer	0	1	2	3	4	5	6	7	8
	Fisher	1.8	11	13	11	15	15	22	25	40

TABLE 5.3: Evolution of Fisher ratio across layers for the scattering concentration network S_C with a ReLU with threshold ρ_{rt}, on the CIFAR dataset.

Learned scattering projections. Beyond scattering trees, the projections P_j of a scattering transform (5.7) can be redefined as arbitrary orthogonal 1×1 convolutional operators, which reduce the number of scattering channels: $P_j P_j^{\mathrm{T}} = \mathrm{Id}$. Orthogonal projectors acting along the direction index ℓ of wavelet filters can improve classification (Oyallon and Mallat, 2015). We are now going to learn these linear operators together with the final linear classifier. Before computing this projection, the mean and variances of each scattering channel is standardized with a batch normalization B_N, by setting affine coefficients $\gamma = 1$ and $\beta = 0$. This projected scattering operator can be written

$$S_P = \prod_{j=1}^{J} P_j \, B_N \, \rho_r \, W.$$

Applying a linear classifier to $S_P(x)$ defines a deep convolutional network whose parameters are the 1×1 convolutional P_j and the classifier weights θ, b. The wavelet convolution filters in W are not learned. The orthogonality of P_j is imposed through the gradient steps (5.6) applied to $D = P_j$. Table 5.2 shows that learning the projectors P_j more than halves the scattering classification error of S_P relatively to S_T on CIFAR-10 and ImageNet, reaching AlexNet accuracy on ImageNet, while achieving a higher Fisher ratio.

The learned orthogonal projections P_j create invariants to families of linear transformations along scattering channels that depend upon scales, directions and phases. They correspond to image transformations which have been linearized by the scattering transform. Small diffeomorphisms which deform the image are examples of operators which are linearized by a scattering transform (Mallat, 2012). The learned projector eliminates within-class variabilities which are not discriminative across classes. Since it is linear, it does not improve linear separability or the Fisher ratio. It takes advantage of the non-linear separation produced by the previous scattering layers.

The operator P_j is a projection on a family of orthogonal directions which define new scattering channels, and is followed by a wavelet convolution W along spatial variables. It defines separable convolutional filters WP_j along space and channels. Learning P_j amounts to choosing orthogonal directions so that $\rho_r W P_j$ optimizes the class means separation. If the class distributions are invariant by rotations, the separation can be achieved with wavelet convolutions along the direction index ℓ (Oyallon and Mallat, 2015), but better results are obtained by learning these filters. This separable scattering architecture is different from separable approximations of deep network filters in discrete cosine bases (Ulicny et al., 2019) or in Fourier-Bessel bases (Qiu et al., 2018). A wavelet scattering computes $\rho_r W P_j$ as opposed to a separable decomposition $\rho_r P_j W$, so the ReLU is applied in a higher dimensional space indexed by wavelet variables produced by W. It provides explicit coordinates to analyze the mathematical properties, but it also increase the number of learned parameters as shown in Table D.1, Appendix D.3.

Concentration along scattering channels. A projected scattering transform can separate class means, but does not concentrate class variabilities. To further reduce classification errors, following Section 5.2.1, a concentration is computed with a tight frame soft-thresholding $D_j \rho_t D_j^{\mathrm{T}}$, applied on scattering channels. It increases the dimension of scattering channels with a 1×1 convolutional tight frame D_j^{T}, applies a soft-thresholding ρ_t, and reduces the number

of channels with the 1×1 convolutional operator D_j. The resulting concentrated scattering operator is

$$S_C = \prod_{j=1}^{J} (D_j \, \rho_t \, D_j^{\mathrm{T}}) \, (P_j \, B_N \, \rho_r W). \tag{5.8}$$

It has $2J$ layers, with odd layers computed by separating means with a ReLu ρ_r and even layers computed by concentrating class variabilities with a soft-thresholding ρ_t. According to Section 5.2.1 the soft-threshold is $\lambda = 1.5\sqrt{d/p}$. This soft-thresholding may be replaced by a thresholded rectifier $\rho_{rt}(u) = \max(u - \lambda, 0)$ with a lower threshold $\lambda = \sqrt{d/p}$. A logistic classifier is applied to $S_C(x)$. The resulting deep network does not include any learned bias parameter, except in the final linear classification layer. Learning is reduced to the 1×1 convolutional operators P_j and D_j along scattering channels, and the linear classification parameters.

Table 5.2 gives the classification errors of this concentrated scattering on CIFAR for $J = 4$ (8 layers) and ImageNet for $J = 6$ (12 layers). The layer dimensions are specified in Appendix D.3. The number of parameters of the scattering networks are given in Table D.1, Appendix D.3. This concentration step reduces the error of S_C by about 40% relatively to a projected scattering S_P. A ReLU thresholding ρ_{rt} produces an error slightly below a soft-thresholding ρ_t both on CIFAR-10 and ImageNet, and this error is also below the errors of ResNet-20 for CIFAR and ResNet-18 for ImageNet. These errors are also nearly half the classification errors previously obtained by cascading a scattering tree S_T with several 1×1 convolutional layers and large MLP classifiers (Zarka et al., 2020; Oyallon et al., 2017). It shows that the separation and concentration learning must be done at each scale rather than at the largest scale output. Table 5.3 shows the progressive improvement of the Fisher ratio measured at each layer of S_C on CIFAR-10. The transition from an odd layer $2j - 1$ to an even layer $2j$ results from $D_j \rho_t D_j^{\mathrm{T}}$, which always improve the Fisher ratio by concentrating class variabilities. The transition from $2j$ to $2j + 1$ is done by $P_{j+1} \rho_r W$, which may decrease the Fisher ratio because of the projection P_{j+1}, but globally brings an important improvement.

5.4 Discussion

We proved that separation and concentration of probability measures can be achieved with rectifiers and thresholdings applied to appropriate tight frames. We also showed that the separation of class means can be achieved by cascading wavelet frames that are not learned. It defines a scattering transform. By concentrating variabilities with a thresholding along scattering channels, we reach ResNet-18 classification accuracy on CIFAR-10 and ImageNet.

These results are refined in Chapter 6, which shows that separation operators are both necessary and sufficient to reach ResNet-18 accuracy. These separation operators can further be restricted to phase collapses of wavelet coefficients. Their separation properties do not come from linear invertibility, which is a weak condition, but from collapsing *multiplicative* within-class variability coming from small deformations.

A major mathematical issue is to understand the mathematical properties of the learned projectors and tight frames along scattering channels. This is necessary to understand the types of classification problems that are well approximated with such architectures. We present results towards this goal in Chapter 7.

Chapter

6 Phase Collapse in Deep Networks

Chapter content

6.1 Introduction **83**
6.2 Eliminating spatial variability with phase collapses **85**
6.3 Learned scattering network with phase collapses **86**
6.4 Phase collapses versus amplitude reductions **88**
6.5 Iterating phase collapses and amplitude reductions **91**
6.5.1 Iterated phase collapses 91
6.5.2 Iterated amplitude reductions 92
6.6 Discussion **93**

We have introduced in Chapter 5 two different types of operators to linearly separate image classes and concentrate their variability. In this chapter, we propose more constrained separation operators which collapse the phases of wavelet coefficients. These operators also concentrate intra-class variability arising from small deformations. It raises the question whether increases in classification accuracy of deep networks which iterate ReLUs with biases results from phase collapses or thresholding operators that improve discrimination through sparsity.

This chapter demonstrates that collapsing the phases of complex wavelet coefficients is sufficient to reach the classification accuracy of ResNets of similar depths. However, replacing the phase collapses with thresholding operators that enforce sparsity considerably degrades the performance. We explain these numerical results by showing that the iteration of phase collapses progressively improves separation of classes, as opposed to thresholding non-linearities.

This chapter is adapted from the following publication: Florentin Guth, John Zarka, and Stéphane Mallat. Phase collapse in neural networks. In *International Conference on Learning Representations*, 2022. We note that the theoretical and numerical results presented in this chapter have been refined by Zarka (2022), who notably presents generalizations of Theorem 6.1 to deformations rather than translations, building on Mallat (2012).

6.1 Introduction

CNN image classifiers progressively eliminate spatial variables through iterated filterings and subsamplings, while linear classification accuracy improves as depth increases (Oyallon, 2017). It has also been numerically observed that CNNs concentrate training samples of each class in small separated regions of a progressively lower-dimensional space. It can ultimately produce a *neural collapse* (Papyan et al., 2020), where all training samples of each class are mapped to a single point. In this case, the elimination of spatial variables comes with a collapse of within-class variability and perfect linear separability. This increase in linear classification accuracy is obtained in standard CNN architectures like ResNets from the iteration of linear convolutional operators and ReLUs with biases.

A difficulty in understanding the underlying mathematics comes from the flexibility of ReLUs. Indeed, a linear combination of biased ReLUs can approximate any non-linearity. Many papers interpret iterations on ReLUs and linear operators as sparse code computations (Sun et al., 2018; Sulam et al., 2018, 2019; Mahdizadehaghdam et al., 2019; Zarka et al., 2020). We show that it is a different mechanism, called *phase collapse*, which underlies the increase in classification accuracy of these architectures. A phase collapse is the elimination of phases of complex-valued wavelet coefficients with a modulus, which we show to concentrate spatial variability. This is demonstrated by introducing a structured convolutional neural network with wavelet filters and no biases.

Section 6.2 introduces and explains phase collapses. Complex-valued representations are used because they reveal the mathematics of spatial variability. Indeed, translations are diagonalized in the Fourier basis, where they become a complex phase shift. Invariants to translations are computed with a modulus, which collapses the phases of this complex representation. Section 6.2 explains how this can improve linear classification. Phase collapses can also be calculated with ReLUs and real filters. A CNN with complex-valued filters is indeed just a particular instance of a real-valued CNN, whose channels are paired together to define complex numbers.

Section 6.3 demonstrates the role of phase collapse in deep classification architectures. It introduces a Learned Scattering network with phase collapses. This network applies a learned 1×1 convolutional complex operator P_j on each layer x_j, followed by a phase collapse, which is obtained with a complex wavelet filtering operator W and a modulus

$$x_{j+1} = |W P_j x_j|. \tag{6.1}$$

It does not use any bias. This network architecture is illustrated in Figure 6.1. With the addition of skip-connections, we show that this phase collapse network reaches ResNet accuracy on ImageNet and CIFAR-10.

Section 6.4 compares phase collapses with other non-linearities such as thresholdings or more general amplitude reduction operators. Such non-linearities can enforce sparsity but do not modify the phase. We show that the accuracy of a Learned Scattering network is considerably reduced when the phase collapse modulus is replaced by soft-thresholdings with learned biases. This is also true of more general phase-preserving non-linearities and architectures.

Section 6.5 explains the performance of iterated phase collapses by showing that each phase collapse progressively improves linear discriminability. On the opposite, the improvements in classification accuracy of successive sparse code computations are shown to quickly saturate.

The main contribution of this chapter is a demonstration that the classification accuracy of deep neural networks mostly relies on phase collapses, which are sufficient to linearly separate the different classes on natural image databases. This is captured by the Learned Scattering architecture which reaches ResNet-18 accuracy on ImageNet and CIFAR-10. We also show that phase collapses are necessary to reach this accuracy, by demonstrating numerically and theoretically that iterating phase-preserving non-linearities leads to a significantly worse performance.

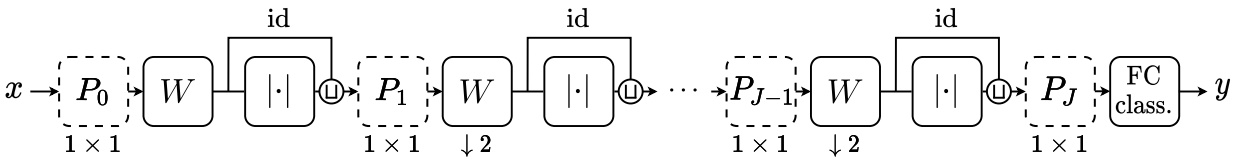

FIGURE 6.1: Architecture of a Learned Scattering network with phase collapses. It has $J + 1$ layers with $J = 11$ for ImageNet and $J = 8$ for CIFAR-10. Each layer is computed with a 1×1 convolutional operator P_j which linearly combines channels. It is followed by a phase collapse, computed with a spatial convolutional filtering with a complex wavelet W and a complex modulus $|\cdot|$. A layer of depth j corresponds to a scale $2^{j/2}$ and a subsampling by 2 is applied every two layers, after W. A skip-connection concatenates the outputs of WP_j and $|WP_j|$. A final 1×1 P_J reduces the dimension before a linear classifier.

6.2 Eliminating spatial variability with phase collapses

Deep convolutional classifiers achieve linear separation of image classes. We show that linear classification on raw images has a poor accuracy because image classes are invariant to local translations. This geometric within-class variability takes the form of random phase fluctuations, and as a result all classes have a zero mean. To improve classification accuracy, non-linear operators must separate class means, which therefore requires to collapse these phase fluctuations.

Translations and phase shifts. Translations capture the spatial topology of the grid on which the image is defined. These translations are transformed into phase shifts by a Fourier transform. We prove that this remains approximately valid for images convolved with appropriate complex filters.

Let x be an image indexed by $u \in \mathbb{Z}^2$. We write $x_\tau(u) = x(u - \tau)$ the translation of x by τ. It is diagonalized by the Fourier transform $\hat{x}(\omega) = \sum_u x(u)\, e^{-i\omega \cdot u}$, which creates a phase shift

$$\hat{x}_\tau(\omega) = e^{-i\omega \cdot \tau} \hat{x}(\omega). \tag{6.2}$$

This diagonalization explains the need to introduce complex numbers to analyze the mathematical properties of geometric within-class variabilities. Computations can however be carried with real numbers, as we will show.

A Fourier transform is computed by filtering x with complex exponentials $e^{i\omega \cdot u}$. One may replace these by complex wavelet filters ψ that are localized in space and in the Fourier domain. The following theorem proves that small translations can still be approximated by a phase shift in this case. We denote by $*$ the convolution of images.

Theorem 6.1. *Let $\psi \colon \mathbb{Z}^2 \to \mathbb{C}$ be a filter with $\|\psi\|_2 = 1$, whose center frequency ξ and bandwidth σ are defined by*

$$\xi = \frac{1}{(2\pi)^2} \int_{[-\pi,\pi]^2} \omega\, |\hat{\psi}(\omega)|^2 \, \mathrm{d}\omega \quad \textit{and} \quad \sigma^2 = \frac{1}{(2\pi)^2} \int_{[-\pi,\pi]^2} |\omega - \xi|^2 |\hat{\psi}(\omega)|^2 \, \mathrm{d}\omega.$$

Then, for any $\tau \in \mathbb{Z}^2$,

$$\|x_\tau * \psi - e^{-i\xi \cdot \tau}(x * \psi)\|_\infty \leq \sigma\, |\tau|\, \|x\|_2. \tag{6.3}$$

The proof is in Appendix E.1. This theorem proves that if $|\tau| \ll 1/\sigma$ then $x_\tau * \psi \approx e^{-i\xi \cdot \tau} x * \psi$. In this case, a translation by τ produces a phase shift by $\xi \cdot \tau$.

Phase collapse and stationarity. We define a *phase collapse* as the elimination of the phase created by a spatial filtering with a complex wavelet ψ. We now show that phase collapses improve linear classification of classes that are invariant to global or local translations.

The training images corresponding to the class label y may be represented as the realizations of a random vector x_y. To achieve linear separation, it is sufficient that class means $\mathbb{E}[x_y]$ are separated and within-class variances around these means are small enough (Hastie et al., 2009). The goal of classification is to find a representation of the input images in which these properties hold.

To simplify the analysis, we consider the particular case where each class y is invariant to translations. More precisely, each random vector x_y is stationary, which means that its probability distribution is invariant to translations. Equation (6.2) then implies that the phases of Fourier coefficients of x_y are uniformly distributed in $[0, 2\pi]$, leading to $\mathbb{E}[\hat{x}_y(\omega)] = 0$ for $\omega \neq 0$. The class means $\mathbb{E}[x_y]$ are thus constant images whose pixel values are all equal to $\mathbb{E}[\hat{x}_y(0)]$. A linear classifier can then only rely on the average colors of the classes, which are often equal in practice. It thus cannot discriminate such translation-invariant classes.

Eliminating uniform phase fluctuations of non-zero frequencies is thus necessary to create separated class means, which can be achieved with the modulus of the Fourier transform. It is a translation-invariant representation: $|\hat{x}_\tau| = |\hat{x}|$. This improves linear discriminability of stationary classes, because $\mathbb{E}[|\hat{x}_y|]$ may be different for different y. However, $|\hat{x}_y|$ has a high variance, because the Fourier transform is unstable to small deformations (Bruna and Mallat, 2013).

Fourier modulus descriptors can be improved by using filters ψ that have a localized support in space. Theorem 6.1 shows that the phase of $x_y * \psi$ is also uniformly distributed in $[0, 2\pi]$. It results that $\mathbb{E}[x_y * \psi] = 0$, and $x * \psi$ still provides no information for linear classification. Applying a modulus similarly computes approximate invariants to small translations: $|x_\tau * \psi| \approx |x * \psi|$, with an error bounded by $\sigma\,|\tau|\,\|x\|_2$. More generally, these *phase collapses* compute approximate invariants to deformations which are well approximated by translations over the support of ψ. This representation improves linear classification by creating different non-zero class means $\mathbb{E}[|x_y * \psi|]$ while achieving a lower variance than Fourier coefficients, as it is stable to deformations (Bruna and Mallat, 2013).

Image classes are usually not invariant to global translations, because of e.g. centered subjects or the sky located in the topmost part of the image. However, classes are often invariant to local translations, up to an unknown maximum scale. This is captured by the notion of local stationarity, which means that the probability distribution of x_y is nearly invariant to translations smaller than some maximum scale (Priestley, 1965). The above discussion remains valid if x_y is only locally stationary over a domain larger than the support of ψ. The use of so-called "windowed absolute spectra" $\mathbb{E}[|x_y * \psi|]$ for locally stationary processes has previously been studied in Tygert et al. (2016).

Real or complex networks. The use of complex numbers is a mathematical abstraction which allows diagonalizing translations, which are then represented by complex phases. It provides a mathematical interpretation of filtering operations performed on real numbers. We show that a real network can still implement complex phase collapses.

In the first layer of a CNN, one can observe that filters are often oscillatory patterns with small supports, where some filters have nearly the same orientation and frequency but with a phase shifted by some α (Krizhevsky et al., 2012). We reproduce in Figure 6.2 a figure from Shang et al. (2016) which evidences this phenomenon. It shows that real filters may be arranged in groups $(\psi_\alpha)_\alpha$ that can be written $\psi_\alpha = \mathrm{Re}(e^{-i\alpha}\psi)$ for a single complex filter ψ and several phases α. This suggests that real-valued networks may indeed implement phase collapses using eq. (6.4). A CNN with complex filters is thus a structured real-valued CNN, where several real filters $(\psi_\alpha)_\alpha$ have been regrouped into a single complex filter ψ. This structure simplifies the mathematical interpretation of non-linearities by explicitly defining the phase, which is otherwise a hidden variable relating multiple filter outputs within each layer.

A phase collapse is explicitly computed with a complex wavelet filter and a modulus. It can also be implicitly calculated by real-valued CNNs. Indeed, for any real-valued signal x, we have

$$|x * \psi| = \frac{1}{2}\int_{-\pi}^{\pi} \mathrm{ReLU}(x * \psi_\alpha)\,\mathrm{d}\alpha. \tag{6.4}$$

Furthermore, this integral is well approximated by a sum over 4 phases, allowing to compute complex moduli with real-valued filters and ReLUs without biases. See Appendix E.2 for a proof of eq. (6.4) and its approximation.

6.3 Learned scattering network with phase collapses

This section introduces a learned scattering transform, which is a highly structured CNN architecture relying on phase collapses and reaching ResNet accuracy on the ImageNet (Russakovsky

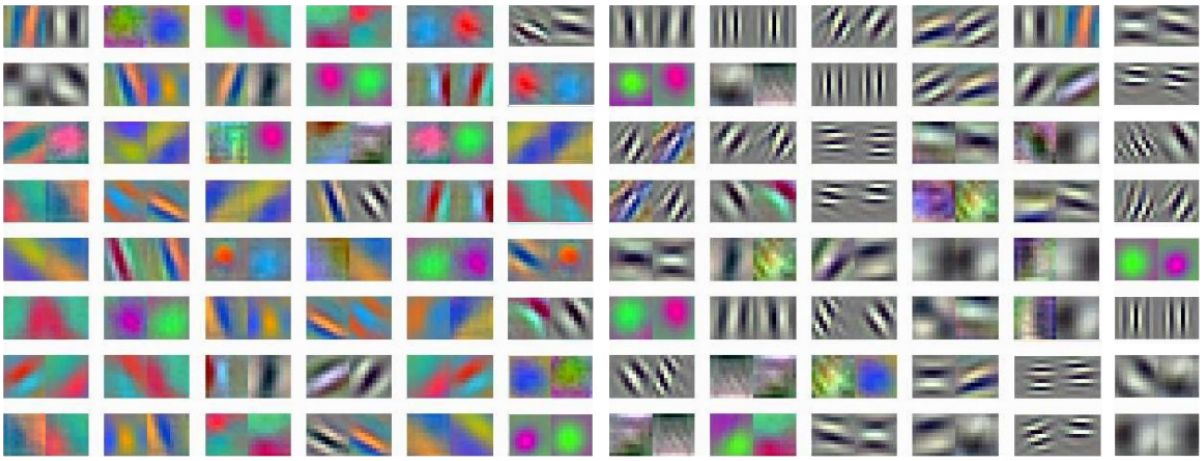

FIGURE 6.2: *First-layer filters from AlexNet (Krizhevsky et al., 2012). They have been paired so that they approximately correspond to two different phases of the same complex filter ψ. Figure reproduced from Shang et al. (2016).*

et al., 2015) and CIFAR-10 (Krizhevsky, 2009) datasets.

Scattering transform. Theorem 6.1 proves that a modulus applied to the output of a complex wavelet filter produces a locally invariant descriptor. This descriptor can then be subsampled, depending upon the filter's bandwidth. We briefly review the scattering transform (Mallat, 2012; Bruna and Mallat, 2013), which iterates phase collapses.

A scattering transform over J scales is implemented with a network of depth J, whose filters are specified by the choice of wavelet. Let $x_0 = x$. For $0 \leq j < J$, the $(j+1)$-th layer x_{j+1} is computed by applying a phase collapse on the j-th layer x_j. It is implemented by a modulus which collapses the phases created by a wavelet filtering operator W:

$$x_{j+1} = |W x_j|. \tag{6.5}$$

The operator W is defined with Morlet filters (Bruna and Mallat, 2013). It has one low-pass filter g_0, and L zero-mean complex band-pass filters $(g_\ell)_\ell$, having an angular direction $\ell\pi/L$ for $0 < \ell \leq L$. It thus transforms an input image $x(u)$ into $L+1$ sub-band images which are subsampled by 2:

$$W x(u, \ell) = x * g_\ell(2u). \tag{6.6}$$

The cascade of j low-pass filters g_0 with a final band-pass filter g_ℓ, each followed by a subsampling, computes wavelet coefficients at a scale 2^j. One can also modify the wavelet filtering W to compute intermediate scales $2^{j/2}$, as explained in Appendix E.5. The spatial subsampling is then only computed every other layer, and the depth of the network becomes twice larger. Applying a linear classifier on such a scattering transform gives good results on simple classification problems such as MNIST (LeCun et al., 2010). However, results are well below ResNet accuracy on CIFAR-10 and ImageNet, as shown in Table 6.1.

Learned Scattering. We have showed in Chapter 5 that a scattering transform can reach ResNet accuracy by incorporating learned 1×1 convolutional operators and soft-thresholding non-linearities in-between wavelet filters. In contrast, we now introduce a Learned Scattering architecture whose sole non-linearity is a phase collapse. It shows that neither biases nor thresholdings are necessary to reach a high accuracy in image classification. A similar result had previously been obtained on image denoising (Mohan et al., 2019).

The Learned Scattering (LScat) network inserts in eq. (6.5) a learned complex 1×1 convolutional operator P_j which reduces the channel dimensionality of each layer x_j before each phase collapse:

$$x_{j+1} = |W P_j x_j|. \tag{6.7}$$

Similar architectures which separate space-mixing and channel-mixing operators had previously been studied in the context of basis expansion (Qiu et al., 2018; Ulicny et al., 2019) or to filter scattering channels (Cotter and Kingsbury, 2019). This separation is also a major feature of recent architectures such as Vision Transformers (Dosovitskiy et al., 2021) or MLP-Mixer (Tolstikhin et al., 2021).

Each P_j computes discriminative channels whose spatial variability is eliminated by the phase collapse operator. Their role is further discussed in Section 6.5. Table 6.1 gives the accuracy of a linear classifier applied to the last layer of this Learned Scattering. It provides an important improvement over a scattering transform, but it does not yet reach the accuracy of ResNet-18.

Including the linear classifier, the architecture uses a total number of layers $J + 1 = 12$ for ImageNet and $J+1 = 9$ for CIFAR, by introducing intermediate scales. The number of channels of $P_j x_j$ is the same as in a standard ResNet architecture (He et al., 2016) and remains no larger than 512. More details are provided in Appendix E.5.

Skip-connections across moduli. Equation (6.7) imposes that all phases are collapsed at each layer, after computing a wavelet transform. More flexibility is provided by adding a skip-connection which concatenates $W P_j x_j$ with its modulus:

$$x_{j+1} = \left[|W P_j x_j| \, , \, W P_j x_j \right]. \tag{6.8}$$

The skip-connection produces a cascade of convolutional filters W without non-linearities in-between. The resulting convolutional operator $WW \cdots W$ is a "wavelet packet" transform which generalizes the wavelet transform (Coifman and Wickerhauser, 1992). Wavelet packets are obtained as the cascade of low-pass and band-pass filters $(g_\ell)_\ell$, each followed by a subsampling. Besides wavelets, wavelet packets include filters having a larger spatial support and a narrower Fourier bandwidth. A wavelet packet transform is then similar to a local Fourier transform. Applying a modulus on such wavelet packet coefficients defines local spatial invariants over larger domains.

As discussed in Section 6.2, image classes are usually invariant to local rather than global translations. Section 6.2 explains that a phase collapse improves discriminability for image classes that are locally translation-invariant over the filter's support. Indeed, phases of wavelet coefficients are then uniformly distributed over $[0, 2\pi]$, yielding zero-mean coefficients for all classes. At scales where there is no local translation-invariance, these phases are no longer uniformly distributed, and they encode information about the spatial localization of features. Introducing a skip-connection provides the flexibility to choose whether to eliminate phases at different scales or to propagate them up to the last layer. Indeed, the next 1×1 operator P_{j+1} linearly combines $|W P_j x_j|$ and $W P_j x_j$ and may learn to use only one of these. This adds some localization information, which appears to be important.

Table 6.1 shows that the skip-connection indeed improves classification accuracy. A linear classifier on this Learned Scattering reaches ResNet-18 accuracy on CIFAR-10 and ImageNet. It demonstrates that collapsing appropriate phases is sufficient to obtain a high accuracy on large-scale classification problems. Learning is reduced to 1×1 convolutions $(P_j)_j$ across channels.

6.4 Phase collapses versus amplitude reductions

We now compare phase collapses with amplitude reductions, which are non-linearities which preserve the phase and act on the amplitude. We show that the accuracy of a Learned Scat-

		Scat	LScat	LScat + skip	ResNet
CIFAR-10	Top-1 error (%)	27.7	11.7	7.7	8.8
ImageNet	Top-5 error (%)	54.1	15.2	11.0	10.9
	Top-1 error (%)	73.0	35.9	30.1	30.2

TABLE 6.1: Error of linear classifiers applied to a scattering (Scat), learned scattering (LScat) and learned scattering with skip connections (+ skip), on CIFAR-10 and ImageNet. The last column gives the single-crop error of ResNet-20 for CIFAR-10 and ResNet-18 for ImageNet, taken from https://pytorch.org/vision/stable/models.html.

tering network is considerably reduced when the phase collapse modulus is replaced by soft-thresholdings with learned biases. This result remains true for other amplitude reductions and architectures.

Thresholding and sparsity. A complex soft-thresholding reduces the amplitude of its input $z = |z|e^{i\varphi}$ by b while preserving the phase: $\rho_b(z) = \mathrm{ReLU}(|z| - b)\, e^{i\varphi}$. Similarly to its real counterpart, it is obtained as the proximal operator of the complex modulus (Yang et al., 2012)

$$\rho_b(z) = \underset{w \in \mathbb{C}}{\arg\min}\ b|w| + \frac{1}{2}|w - z|^2. \tag{6.9}$$

Soft-thresholdings and moduli have opposite properties, since soft-thresholdings preserve the phase while attenuating the amplitude, whereas moduli preserve the amplitude while eliminating the phase. In contrast, ReLUs with biases are more general non-linearities which can act both on phase and amplitude. This is best illustrated over $\mathbb{R}$ where the phase is replaced by the sign, through the even-odd decomposition. If $z \in \mathbb{R}$ and $\lambda \geq 0$, then the even part of $\mathrm{ReLU}(z - \lambda)$ is $\mathrm{ReLU}(|z| - \lambda)$, which is an absolute value with a dead-zone $[-\lambda, \lambda]$. When $\lambda = 0$, it becomes an absolute value $|z|$. The odd part is a soft-thresholding $\rho_\lambda(z) = \mathrm{sign}(z)\,\mathrm{ReLU}(|z| - \lambda)$. Over $\mathbb{C}$, a similar result can be obtained through the decomposition into phase harmonics (Mallat et al., 2019).

We have explained how phase collapses can improve the classification accuracy of locally stationary processes by separating class means $\mathbb{E}[|x_y * \psi|]$. In contrast, since the phase of $x_y * \psi$ is uniformly distributed for such processes, then it is also true of $\rho_\lambda(x_y * \psi)$. This implies that $\mathbb{E}[\rho_\lambda(x_y * \psi)] = 0$ for all λ. Class means of locally stationary processes are thus not separated by a thresholding.

When class means $\mathbb{E}[x_y * \psi]$ are separated, a soft-thresholding of $x_y * \psi$ may however improve classification accuracy. If $x_y * \psi$ is sparse, then a soft-thresholding $\rho_\lambda(x_y * \psi)$ reduces the within-class variance as shown in Chapter 5. Coefficients below the threshold may be assimilated to unnecessary "clutter" which is set to 0. To improve classification, convolutional filters must then produce high-amplitude coefficients corresponding to discriminative "features".

Phase collapses versus amplitude reductions. A Learned Scattering with phase collapses preserves the amplitudes of wavelet coefficients and eliminates their phases. On the opposite, one may use a non-linearity which preserves the phases of wavelet coefficients but attenuates their amplitudes, such as a soft-thresholding. We show that such non-linearities considerably degrade the classification accuracy compared to phase collapses.

Several previous works made the hypothesis that sparsifying neural responses with thresholdings is a major mechanism for improving classification accuracy (Sun et al., 2018; Sulam et al., 2018, 2019; Mahdizadehaghdam et al., 2019; Zarka et al., 2020). The dimensionality of

	Scat	LScat				
		Mod	AThresh	ATanh	ASigmoid	ASign
Without skip	27.7	11.7	36.7	40.7	38.5	39.9
With skip	-	7.7	22.5	19.2	17.0	19.5

Table 6.2: Top-1 error (in %) on CIFAR-10 with a linear classifier applied to a Scattering network (Scat) and several Learned Scattering networks (LScat) with several non-linearities. They include a modulus (Mod), an amplitude soft-thresholding (Thresh), an amplitude hyperbolic tangent (ATanh), an amplitude sigmoid (ASigmoid), and an amplitude Soft-sign (ASign).

sparse representations can then be reduced with random filters which implement a form of compressed sensing (Donoho, 2006; Candes et al., 2006). The interpretation of CNNs as compressed sensing machines with random filters has been studied (Giryes et al., 2016), but it never led to classification results close to e.g. ResNet accuracy.

To test this hypothesis, we replace the modulus non-linearity in the Learned Scattering architecture with thresholdings, or more general phase-preserving non-linearities. A Learned Amplitude Reduction Scattering applies a non-linearity $\rho(z)$ which preserves the phases of wavelet coefficients $z = |z|e^{i\varphi}$: $\rho(z) = e^{i\varphi}\,\rho(|z|)$. Without skip-connections, each layer x_{j+1} is computed from x_j with

$$x_{j+1} = \rho(WP_j x_j), \tag{6.10}$$

and with skip-connections

$$x_{j+1} = \Big[\rho(WP_j x_j)\,,\, WP_j x_j\Big]. \tag{6.11}$$

A soft-thresholding is defined by $\rho(|z|) = \mathrm{ReLU}(|z| - \lambda)$ for some threshold λ. We also define an amplitude hyperbolic tangent $\rho(|z|) = (e^{|z|} - e^{-|z|})/(e^{|z|} + e^{-|z|})$, an amplitude sigmoid as $\rho(|z|) = (1 + e^{-\gamma \log|z| - \lambda})^{-1}$ and an amplitude soft-sign as $\rho(|z|) = |z|/(1 + |z|)$. The soft-thresholding and sigmoid parameters γ and λ are learned for each layer and each channel.

We evaluate the classification performance of a Learned Amplitude Reduction Scattering on CIFAR-10, by applying a linear classifier on the last layer. Classification results are given in Table 6.2 for different amplitude reductions, with or without skip-connections. Learned Amplitude Reduction Scatterings yield much larger errors than a Learned Scattering with phase collapses. Without skip-connections, they are even above a scattering transform, which also uses phase collapses but does not have learned 1×1 convolutional projections $(P_j)_j$. It demonstrates that high accuracies result from phase collapses without biases, as opposed to amplitude reduction operators including thresholdings, which learn bias parameters.

We repeat this comparison in the real domain, using a standard ResNet-18 architecture without biases. We replace the ReLU non-linearity by an absolute value or sign collapse $|x|$ and several sign-preserving (i.e., odd) non-linearities. They include a soft-thresholding $\rho_\lambda(x) = \mathrm{sign}(x)\,\mathrm{ReLU}(|x| - \lambda)$, an hyperbolic tangent $\rho(x) = (e^x - e^{-x})/(e^x + e^{-x})$, and a soft-sign $\rho(x) = x/(1 + |x|)$. We do not report results for an amplitude sigmoid $\rho(x) = \mathrm{sign}(x)(1 + e^{-\gamma \log|x| - \lambda})^{-1}$ because of optimization instabilities when learning the parameters γ and λ. Classification results on the ImageNet dataset are given in Table 6.3. The error of bias-free ReLUs and sign collapses are comparable to a standard ResNet-18, and confirm that sign collapses are sufficient to reach such accuracies. In contrast, the performance of amplitude reduction non-linearities, which preserve the sign of network coefficients, is significantly worse. The superiority of phase collapses on amplitude reductions thus still holds in the real domain and when the spatial filters are not constrained to be wavelets.

	ResNet	BFResNet				
		ReLU	Abs	Thresh	Tanh	Sign
Top-5 error (%)	10.9	12.3	13.9	25.7	22.4	24.2
Top-1 error (%)	30.2	32.6	35.3	50.0	44.6	49.3

Table 6.3: Classification errors on ImageNet of bias-free ResNet-18 (BFResNet) architectures with several non-linearities. They include a ReLU, an absolute value which performs sign collapses (Abs), a soft-thresholding (Thresh), a hyperbolic tangent (Tanh), and a soft-sign (Sign). They are compared to the original ResNet-18 architecture, which uses a ReLU and learns biases.

ReLUs with biases. Most CNNs, including ResNets, use ReLUs with biases. A ReLU with bias simultaneously affects the sign and the amplitude of its real input. Over complex numbers, it amounts to transforming the phase and the amplitude. These numerical experiments show that accuracy improvements result from acting on the sign or phase rather than the amplitude. Furthermore, this can be constrained to collapsing the phase of wavelet coefficients while preserving their amplitude.

Several CNN architectures have demonstrated a good classification accuracy with iterated thresholding algorithms, which increase sparsity. However, all these architecture also modified the sign of coefficients by computing *non-negative* sparse codes (Sun et al., 2018; Sulam et al., 2018; Mahdizadehaghdam et al., 2019) or with additional ReLU or modulus layers (Zarka et al., 2020, Chapter 5). It seems that it is the sign or phase collapse of these non-linearities which is responsible for good classification accuracies, as opposed to the calculation of sparse codes through iterated amplitude reductions.

6.5 Iterating phase collapses and amplitude reductions

We now provide a theoretical justification to the above numerical results in simplified mathematical frameworks. This section studies the behavior of phase collapses and amplitude reductions when they are iterated over several layers. It shows that phase collapses benefit from iterations over multiple layers, whereas there is no significant gain in performance when iterating amplitude reductions.

6.5.1 Iterated phase collapses

We explain the role of iterated phase collapses with multiple filters at each layer. Classification accuracy is improved through the creation of additional dimensions to separate class means. The learned projectors $(P_j)_j$ are optimized for this separation.

We consider the classification of stationary processes $x_y \in \mathbb{R}^d$, corresponding to different image classes indexed by y. Given a realization x of x_y, and because of stationarity, the optimal linear classifier is calculated from the empirical mean $1/d \sum_u x(u)$. It computes an optimal linear estimation of $\mathbb{E}[x_y(u)] = \mu_y$. If all classes have the same mean $\mu_y = \mu$, then all linear classifiers fail.

As explained in Section 6.2, linear classification can be improved by computing $(|x * \psi_k|)_k$ for some wavelet filters $(\psi_k)_k$. These phase collapses create additional directions with non-zero means which may separate the classes. If x_y is stationary, then $|x_y * \psi_k|$ remains stationary for any ψ_k. An optimal linear classifier applied to $(|x * \psi_k(u)|)_k$ is thus obtained by a linear combination of all empirical means $(1/d \sum_u |x * \psi_k(u)|)_k$. They are proportional to the ℓ^1 norm $\|x * \psi_k\|_1$, which is a measure of sparsity of $x * \psi_k$.

If linear classification on $(|x * \psi_k(u)|)_k$ fails, it reveals that the means $\mathbb{E}[|x_y * \psi_k(u)|] = \mu_{y,k}$ are not sufficiently different. Separation can be improved by considering the spatial variations

of $|x_y * \psi_k(u)|$ for different y. These variations can be revealed by a phase collapse on a new set of wavelet filters $\psi_{k'}$, which computes $(||x * \psi_k| * \psi_{k'}|)_{k,k'}$. This phase collapse iteration is the principle used by scattering transforms to discriminate textures (Bruna and Mallat, 2013; Sifre and Mallat, 2013): each successive phase collapse creates additional directions to separate class means.

However, this may still not be sufficient to separate class means. More discriminant statistical properties may be obtained by linearly combining $(|x*\psi_k|)_k$ across k before applying a new filter $\psi_{k'}$. In a Learned Scattering with phase collapse, this is done with a linear projector P_1 across the channel indices k, before computing a convolution with the next filter $\psi_{k'}$. The 1×1 operator P_1 is optimized to improve the linear classification accuracy. It amounts to learning weights w_k such that $\mathbb{E}[\left|\sum_k w_k |x_y * \psi_k| * \psi_{k'}\right|]$ is as different as possible for different y. Because these are proportional to the ℓ^1 norms $\left\|\sum_k w_k |x * \psi_k| * \psi_{k'}\right\|_1$, it means that the images $\sum_k w_k |x * \psi_k| * \psi_{k'}$ have different sparsity levels depending upon the class y of x. The weights $(w_k)_k$ of P_1 can thus be interpreted as features along channels providing different sparsifications for different classes. A Learned Scattering network learns such P_j at each scale j.

6.5.2 Iterated amplitude reductions

Sparse representations and amplitude reduction algorithms may improve linear classification by reducing the variance of class mean estimations, which can be interpreted as clutter removal. We studied these approaches in Chapter 5 by modeling the clutter as an additive white noise. Although a single thresholding step may improve linear classification, we show that iterating more than one thresholding does not improve the classification accuracy, if no phase collapses are inserted.

To understand these properties, we consider the discrimination of classes x_y for which class means $\mathbb{E}[x_y] = \mu_y$ are all different. If there exists y' such that $\|\mu_y - \mu_{y'}\|$ is small, then the class y can still be discriminated from y' if we can estimate $\mathbb{E}[x_y]$ sufficiently accurately from a single realization of x_y. This is a mean estimation problem. Suppose that $x_y = \mu_y + \mathcal{N}(0, \sigma^2)$ is contaminated with Gaussian white noise, where the noise models some clutter. Suppose also that there exists a linear orthogonal operator D such that $D^{\mathrm{T}}\mu_y$ is sparse for every y, and hence has its energy concentrated in few non-zero coefficients. Such a D may be computed by minimizing the expected ℓ^1 norm $\sum_y \mathbb{E}\left[\|D^{\mathrm{T}} x_y\|_1\right]$. The estimation of μ_y can be improved with a soft-thresholding estimator (Donoho and Johnstone, 1994), which sets to zero all coefficients below a threshold λ proportional to σ. It amounts to computing $\rho_\lambda(D^{\mathrm{T}}x)$, where ρ_λ is a soft-thresholding.

However, we explain below why this approach cannot be further iterated without inserting phase collapses. The reason is that a sparse representation $\rho_\lambda(D^{\mathrm{T}}x)$ concentrates its entropy in the phases of the coefficients, rather than their amplitude. We then show that such processes cannot be further sparsified, which means that a second thresholding $\rho_{\lambda'}(D'^{\mathrm{T}}\rho_\lambda(D^{\mathrm{T}}x))$ will not reduce further the variance of class mean estimators. This entails that a model of within-class variability relying on amplitude reductions cannot be the sole mechanism behind the performance of deep networks.

Iterating amplitude reductions may however be useful if it is alternated with another non-linearity which partly or fully collapses phases. Reducing the entropy of the phases of $\rho_\lambda(D^{\mathrm{T}}x)$ allows $\rho_{\lambda'}D'^{\mathrm{T}}$ to further sparsify the process and hence further reduce the within-class variability. As mentioned in Section 6.4, this is the case for previous work which used iterated sparsification operators (Sun et al., 2018; Sulam et al., 2018; Mahdizadehaghdam et al., 2019). Indeed, these networks compute non-negative sparse codes where sparsity is enforced with a ReLU, which acts both on phases and amplitudes. Our results shows that the benefit of iterating non-negative sparse coding comes from the sign collapse due to the non-negativity constraint.

We now qualitatively demonstrate these claims with two theorems. We first show that finding the sparsest representation of a random process (i.e., minimizing its ℓ^1 norm) is the same as maximizing a lower bound on the entropy of its phases.

Theorem 6.2. *Let x denote a random vector in $\mathbb{C}^d$ with a probability density p. Let $H(x)$ be the entropy of x with respect to the Lebesgue measure*

$$H(x) = -\int p(x) \log p(x) \, \mathrm{d}x.$$

If $D \in U(d)$ is a unitary operator, then

$$H\Big(\varphi(D^{\mathrm{T}}x) \,\Big|\, \Big|D^{\mathrm{T}}x\Big|\Big) \geq H(x) - d - 2d \log\Big(\frac{1}{d}\mathbb{E}\Big[\Big\|D^{\mathrm{T}}x\Big\|_1\Big]\Big),$$

where $\varphi(D^{\mathrm{T}}x) \in [0, 2\pi]^d$ (resp. $\Big|D^{\mathrm{T}}x\Big| \in \mathbb{R}_+^d$) is the random process of the entry-wise phases (resp. moduli) of $D^{\mathrm{T}}x$.

The proof is in Appendix E.3. This theorem gives a lower-bound on the conditional entropy of the phases of $D^{\mathrm{T}}x$ with a decreasing function of the expected ℓ^1 norm of $D^{\mathrm{T}}x$. Minimizing over D this expected ℓ^1 norm amounts to maximizing the lower bound on $H\Big(\varphi(D^{\mathrm{T}}x) \,\Big|\, \Big|D^{\mathrm{T}}x\Big|\Big)$. An extreme situation arises when this entropy reaches its maximal value of $d\log(2\pi)$. In this case, the phase $\varphi(D^{\mathrm{T}}x)$ has a maximum-entropy distribution and is therefore uniformly distributed in $[0, 2\pi]^d$. Moreover, in this extreme case $\varphi(D^{\mathrm{T}}x)$ is independent from $\Big|D^{\mathrm{T}}x\Big|$, since its conditional distribution does not depend on $\Big|D^{\mathrm{T}}x\Big|$. Such statistical properties have previously been observed on wavelet coefficients of natural images (Wainwright et al., 2001a), where the wavelet transform seems to be a nearly optimal sparsifying unitary dictionary.

The second theorem considers the extreme case of a random process whose phases are conditionally independent and uniform. It proves that such a process cannot be significantly sparsified with a change of basis.

Theorem 6.3. *Assume that $\varphi(\rho_\lambda(D^{\mathrm{T}}x))$ is uniformly distributed in $[0, 2\pi]^d$ and independent from $|\rho_\lambda(D^{\mathrm{T}}x)|$. Then there exists a constant $C_d > 0$ which depends on the dimension d, such that for any $D' \in U(d)$,*

$$\mathbb{E}\Big[\|D'^{\mathrm{T}}\rho_\lambda(D^{\mathrm{T}}x)\|_1\Big] \geq C_d \mathbb{E}\Big[\Big\|\rho_\lambda(D^{\mathrm{T}}x)\Big\|_1\Big].$$

The proof is in Appendix E.4. This theorem shows that random processes with conditionally independent and uniform phases have an ℓ^1 norm which cannot be significantly decreased by any unitary transformation. Numerical evaluations suggest that the constant C_d may be chosen to be $\sqrt{\pi}/2 \approx 0.886$, independently of the dimension d. This constant arises as the value of $\mathbb{E}[|Z|]$ when Z is a complex normal random variable with $\mathbb{E}[|Z|^2] = 1$.

These two theorems explain qualitatively that linear classification on $\rho_\lambda(D^{\mathrm{T}}x)$ cannot be improved by another thresholding that would take advantage of another sparsification operator. Indeed, Theorem 6.2 shows that if $\rho_\lambda(D^{\mathrm{T}}x)$ is sparse, then its phases have random fluctuations of high entropy. Theorem 6.3 indicates that such random phases prevent a further sparsification of $\rho_\lambda(D^{\mathrm{T}}x)$ with some linear operator D'. Applying a second thresholding $\rho_{\lambda'}(D'^{\mathrm{T}}\rho_\lambda(D^{\mathrm{T}}x))$ thus cannot significantly reduce the variance of class mean estimators.

6.6 Discussion

This chapter studies the improvement of linear separability for image classification in deep convolutional networks. We show that it mostly relies on a phase collapse phenomenon. Eliminating the phase of wavelet coefficients improves the separation of class means. We introduced a

Learned Scattering network with wavelet phase collapses and learned 1×1 convolutional filters P_j, which reaches ResNet accuracy. The learned 1×1 operators P_j enhance discriminability by computing channels that have different levels of sparsity for different classes. This architecture is used in Chapter 7 to reduce the study of learned weights to the operators P_j along channels.

When class means are separated, thresholding non-linearities can improve classification by reducing the variance of class mean estimators. When used alone, the classification performance is poor over complex datasets such as ImageNet or CIFAR-10, because class means are not sufficiently separated. Furthermore, the iteration of thresholdings on sparsification operators requires intermediary phase collapses.

These results show that linear separation of classes result from acting on the sign or phase of network coefficients rather than their amplitude. Furthermore, this can be constrained to collapsing the phase of wavelet coefficients while preserving their amplitude. The elimination of spatial variability with phase collapses is thus both necessary and sufficient to linearly separate classes on complex image datasets.

Part III

A Model of Network Weights with Aligned Random Features

Chapter

7 The Rainbow Model of Deep Networks

Chapter content

7.1 Introduction 98
7.2 Rainbow networks 100
7.2.1 Rotations in random feature maps 100
7.2.2 Deep rainbow networks 103
7.2.3 Symmetries and convolutional rainbow networks 109
7.3 Numerical results 111
7.3.1 Convergence of activations in the infinite-width limit 112
7.3.2 Properties of learned weight covariances 114
7.3.3 Gaussian rainbow approximations 120
7.4 Discussion 125

We have introduced in Chapters 5 and 6 constrained non-linear operators to structure deep convolutional network architectures in image classification. It has resulted in learned scattering network architectures, which use fixed spatial wavelet filters but learn linear operators along channels. A major issue is now to understand the nature of these learned operators and their mathematical properties.

This chapter introduce a probabilistic model of the learned weights in deep neural networks. The model cascades random feature maps whose weight distributions are learned. It assumes that dependencies between weights at different layers are reduced to rotations which align the input activations. Neuron weights within a layer are independent after this alignment. Their activations define kernels which become deterministic in the infinite-width limit. This is verified numerically for ResNets trained on the ImageNet dataset.

We also show that the learned weight distributions have low-rank covariances. Rainbow networks thus alternate between linear dimension reductions and non-linear high-dimensional embeddings with white random features. Gaussian rainbow networks are defined with Gaussian weight distributions. These models are validated numerically on image classification on the CIFAR-10 dataset, with wavelet scattering networks. We further show that during training, SGD updates the weight covariances while mostly preserving the Gaussian initialization.

This chapter is adapted from the following preprint: Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *arXiv preprint arXiv:2305.18512*, 2023.

Notations. Some notations of this chapter are reversed with respect to the ones of Chapters 5 and 6, in order to emphasize different aspects of the same operator. In this chapter, we use the letter W to refer to the weight matrix of a linear layer. In Section 7.2.3, we write the operator $W = L\,P$, where L is a learned 1×1 convolution operator (denoted P in Chapters 5 and 6) and P is a predefined spatial convolution (which may be composed of wavelet filters, denoted W in

Chapters 5 and 6). Additionally, for learned scattering architectures, we impose that L and P commute, so that we implement $W = P\,L$. This recovers the order between the operators in Chapter 6.

7.1 Introduction

Deep neural networks have been described as black boxes because many of their fundamental properties are not understood. Their weight matrices are learned by performing stochastic gradient descent from a random initialization. Each training run thus results in a different set of weight matrices, which can be considered as a random realization of some probability distribution. What is this probability distribution? What is the corresponding functional space? Do all networks learn the same function, and even the same weights, up to some symmetries? This chapter addresses these questions.

Theoretical studies have mostly focused on shallow learning. A first line of work has studied learning of the last layer while freezing the other ones. The previous layers thus implement random features (Jarrett et al., 2009; Pinto et al., 2009) which specify a kernel that becomes deterministic in the infinite-width limit (Rahimi and Recht, 2007; Daniely et al., 2016). Learning has then been incorporated in these models. Neal (1996); Williams (1996); Lee et al. (2018); Matthews et al. (2018) show that some networks behave as Gaussian processes. Training is then modeled as sampling from the Bayesian posterior given the training data. On the other hand, Jacot et al. (2018) and Lee et al. (2019b) assume that trained weights have small deviations from their initialization. In these cases, learning is in a "lazy" regime (Chizat et al., 2019) specified by a fixed kernel. It has been opposed to a "rich" or feature-learning regime (Chizat and Bach, 2020; Woodworth et al., 2020), which achieves higher performance on complex tasks (Lee et al., 2020; Geiger et al., 2020). Empirical observations of weight statistics have indeed shown that they significantly evolve during training (Martin and Mahoney, 2021; Thamm et al., 2022). This has been precisely analyzed for one-hidden-layer networks in the infinite-width "mean-field" limit (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020), which allows tracking the neuron weight distribution as it evolves away from the Gaussian initialization during training. The generalization to deeper networks is greatly complicated by the fact that intermediate activations depend on the random weight realizations (Sirignano and Spiliopoulos, 2022; E and Wojtowytsch, 2020; Nguyen and Pham, 2020; Chen et al., 2022c; Yang and Hu, 2021). However, numerical experiments (Raghu et al., 2017; Kornblith et al., 2019) show that intermediate activations correlate significantly across independent realizations, which calls for an explanation of this phenomenon.

Building upon these ideas, we introduce the rainbow model of the joint probability distribution of trained network weights across layers. It assumes that dependencies between the weight matrices W_j at all layers j are reduced to rotations. This means that $W_j = W'_j \hat{A}_{j-1}$, where $W'_1, \dots, W'_J$ are independent random matrices, and $\hat{A}_{j-1}$ is a rotation that depends on the previous layer weights $W_1, \dots, W_{j-1}$. The W'_j are further assumed to be random feature matrices, that is, their rows are independent and identically distributed.

The functional properties of rainbow networks depend on the random feature distribution at each layer. We show numerically that weights of trained networks typically have low-rank covariances. The corresponding rainbow networks thus implement dimensionality reductions in-between the high-dimensional random feature embeddings, similar to previous works (Cho and Saul, 2009; Mairal, 2016; Bietti, 2019). We further demonstrate that input activation covariances provide efficient approximations of the eigenspaces of the weight covariances. The number of model parameters and hence the supervised learning complexity can thus be considerably reduced by unsupervised information.

The weight covariances completely specify the rainbow network output and properties when the weight distributions are Gaussian. The eigenvectors of these weight covariances can be in-

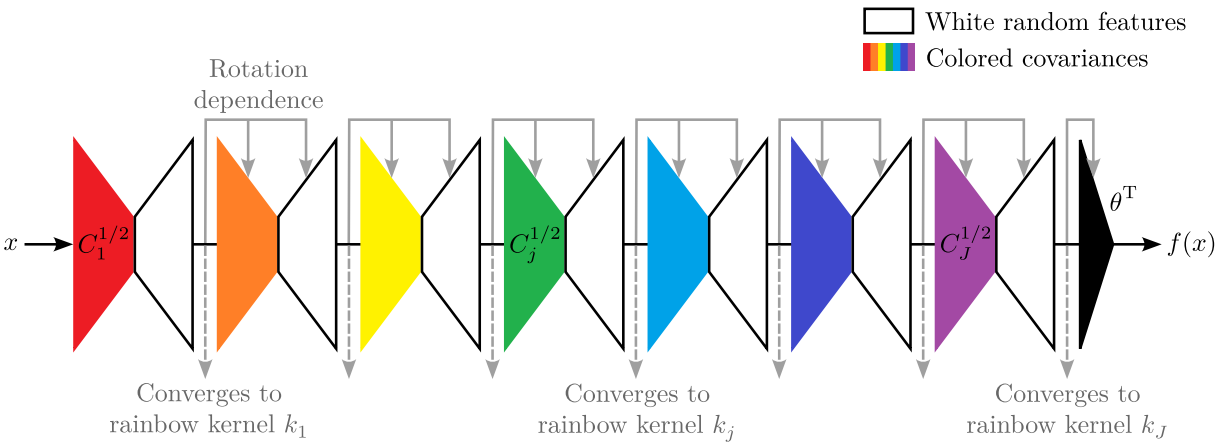

FIGURE 7.1: A deep rainbow network cascades random feature maps whose weight distributions are learned. They typically have a low-rank covariance. Each layer can be factorized into a linear dimensionality reduction determined by the colored covariance, followed by a non-linear high-dimensional embedding with white random features. At each layer, the hidden activations define a kernel which converges to a deterministic rainbow kernel in the infinite width limit. It induces a random rotation of the next layer weights. For Gaussian rainbow networks, the random feature embedding is a dot-product kernel feature map which does not need to be rotated.

terpreted as learned features, rather than individual neuron weights which are random. This Gaussian assumption is too restrictive to model arbitrary trained networks. However, it can approximately hold for architectures which incorporate prior information and restrict their learned weights. In some of our numerical experiments, we will thus consider learned scattering networks introduced in Chapters 5 and 6, which have fixed wavelet spatial filters and learn weights along channels only.

This chapter makes the following main contributions:

- We prove that the rainbow network activations converge to a random rotation of a deterministic kernel feature vector in the infinite-width limit, which explains the empirical results of representation similarity of Raghu et al. (2017) and Kornblith et al. (2019). We verify numerically this convergence on scattering networks and ResNets trained on the CIFAR-10 and ImageNet image classification datasets. We conjecture but do not prove that this convergence conversely implies the first rainbow assumption that layer dependencies are reduced to rotations.

- We validate the Gaussian rainbow model for scattering networks trained on CIFAR-10. We verify that the weight covariances converge up to rotation when the width increases, and that the weights are approximately Gaussian. The weight covariances are sufficient to sample rainbow weights and define new networks that achieve comparable classification accuracy as the original trained network when the width is large enough. Further, we show that SGD training only updates the weight covariances while nearly preserving the white random feature initializations, suggesting a possible explanation for the Gaussian rainbow assumption in this setting.

- We prove that equivariance to general groups can be achieved in rainbow networks with weight distributions that are invariant to the group action. This constraint on distributions rather than on individual neurons (Cohen and Welling, 2016; Kondor and Trivedi, 2018) avoids any weight sharing or synchronizations, which are difficult to implement in biological systems.

The rainbow model is illustrated in Figure 7.1. In Section 7.2, we introduce rainbow networks

and the associated kernels that describe their infinite-width limit. We validate numerically the above properties and results in Section 7.3.

7.2 Rainbow networks

Weight matrices of learned deep networks are strongly dependent across layers. Deep rainbow networks define a mathematical model of these dependencies through rotation matrices that align input activations at each layer. We review in Section 7.2.1 the properties of random features, which are the building blocks of the model. We then introduce in Section 7.2.2 deep fully-connected rainbow networks, which cascade aligned random feature maps. We show in Section 7.2.3 how to incorporate inductive biases in the form of symmetries or local neuron receptive fields. We also extend rainbow models to convolutional networks.

7.2.1 Rotations in random feature maps

We being by reviewing the properties of one-hidden layer random feature networks. We then prove that random weight fluctuations produce a random rotation of the hidden activation layer in the limit of infinite layer width. The rainbow network model will be obtained by applying this result at all layers of a deep network.

Random feature network. A one-hidden layer network computes a hidden activation layer with a matrix W of size $d_1 \times d_0$ and a pointwise non-linearity ρ:

$$\hat{\varphi}(x) = \rho(Wx) \quad \text{for} \quad x \in \mathbb{R}^{d_0}.$$

We consider a random feature network (Rahimi and Recht, 2007). The rows of W, which contain the weights of different neurons, are independent and have the same probability distribution π:

$$W = (w_i)_{i \leq d_1} \quad \text{with i.i.d.} \quad w_i \sim \pi.$$

In many random feature models, each row vector has a known distribution with uncorrelated coefficients (Jarrett et al., 2009; Pinto et al., 2009). Learning is then reduced to calculating the output weights $\hat{\theta}$, which define

$$\hat{f}(x) = \langle \hat{\theta}, \hat{\varphi}(x) \rangle.$$

In contrast, we consider general distributions π which will be estimated from the weights of trained networks in Section 7.3.

Our network does not include any bias for simplicity. Bias-free networks have been shown to achieve comparable performance as networks with biases for denoising (Mohan et al., 2019) and image classification in Chapters 5 and 6. However, biases can easily be incorporated in random feature models and thus rainbow networks.

We consider a normalized network, where ρ includes a division by $\sqrt{d_1}$ so that $\|\hat{\varphi}(x)\|$ remains of the order of unity when the width d_1 increases. We shall leave this normalization implicit to simplify notations, except when illustrating mathematical convergence results. Note that this choice differs from the so-called standard parameterization (Yang and Hu, 2021). In numerical experiments, we perform SGD training with this standard parameterization which avoids getting trapped in the lazy training regime (Chizat et al., 2019). Our normalization convention is only applied at the end of training, where the additional factor of $\sqrt{d_1}$ is absorbed in the next-layer weights $\hat{\theta}$.

We require that the input data has finite energy: $\mathbb{E}_x[\|x\|^2] < +\infty$. We further assume that the non-linearity ρ is Lipschitz, which is verified by many non-linearities used in practice, including ReLU. Finally, we require that the random feature distribution π has finite fourth-order moments.

Kernel convergence. We now review the convergence properties of one-hidden layer random feature networks. This convergence is captured by the convergence of their kernel (Rahimi and Recht, 2007, 2008),

$$\hat{k}(x, x') = \langle \hat{\varphi}(x), \hat{\varphi}(x') \rangle = \frac{1}{d_1} \sum_{i=1}^{d_1} \rho(\langle x, w_i \rangle)\, \rho(\langle x', w_i \rangle),$$

where we have made explicit the factor d_1^{-1} coming from our choice of normalization. Since the rows w_i are independent and identically distributed, the law of large numbers implies that when the width d_1 goes to infinity, this empirical kernel has a mean-square convergence to the asymptotic kernel

$$k(x, x') = \mathbb{E}_{w \sim \pi} \Big[\rho(\langle x, w \rangle)\, \rho(\langle x', w \rangle) \Big]. \tag{7.1}$$

This convergence means that even though $\hat{\varphi}$ is random, its geometry (as described by the resulting kernel) is asymptotically deterministic. As we will see, this imposes that random fluctuations of $\hat{\varphi}(x)$ are reduced to rotations.

Let $\varphi(x)$ be an infinite-dimensional deterministic colored feature vector in a separable Hilbert space H, which satisfies

$$\langle \varphi(x), \varphi(x') \rangle_H = k(x, x'). \tag{7.2}$$

Such feature vectors always exist (Aronszajn, 1950, see also Schölkopf and Smola, 2002). For instance, one can choose $\varphi(x) = (\rho(\langle x, w \rangle))_w$, the infinite-width limit of random features ρW. In that case, $H = L^2(\pi)$, that is, the space of square-integrable functions with respect to π, with dot-product $\langle g, h \rangle_H = \mathbb{E}_{w \sim \pi}[g(w)\, h(w)]$. This choice is however not unique: one can obtain other feature vectors defined in other Hilbert spaces by applying a unitary transformation to φ, which does not modify the dot product in eq. (7.2). In the following, we choose the kernel PCA (KPCA) feature vector, whose covariance matrix $\mathbb{E}_x[\varphi(x)\, \varphi(x)^{\mathrm{T}}]$ is diagonal with decreasing values along the diagonal, introduced by Schölkopf et al. (1997). It is obtained by expressing any feature vector φ in its PCA basis relative to the distribution of x. In this case $H = \ell^2(\mathbb{N})$.

Finally, we denote by $\mathcal{H}$ the reproducing kernel Hilbert space (RKHS) associated to the kernel k in eq. (7.1). It is the space of functions f which can be written $f(x) = \langle \theta, \varphi(x) \rangle_H$, with norm $\|f\|_{\mathcal{H}} = \|\theta\|_H$.[1] A random feature network defines approximations of functions in this RKHS. With $H = L^2(\pi)$, these functions can be written

$$f(x) = \mathbb{E}_{w \sim \pi}[\theta(w)\, \rho(\langle x, w \rangle)] = \int \theta(w)\, \rho(\langle x, w \rangle)\, \mathrm{d}\pi(w).$$

This expression is equivalent to the mean-field limit of one-hidden-layer networks (Chizat and Bach, 2018; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Sirignano and Spiliopoulos, 2020), which we will generalize to deep networks in Section 7.2.2.

Rotation alignment. We now introduce rotations which align approximate kernel feature vectors. By abuse of language, we use rotations as a synonym for orthogonal transformations, and also include improper rotations which are the composition of a rotation with a reflection.

We have seen that the kernel $\hat{k}(x, x') = \langle \hat{\varphi}(x), \hat{\varphi}(x') \rangle$ converges to the kernel $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$. We thus expect (and will later prove) that there exists a rotation $\hat{A}$ such that $\hat{A}\, \hat{\varphi} \approx \varphi$ because all feature vectors of the kernel k are rotations of one another. The rotation $\hat{A}$ is dependent on the random feature realization W and is thus random. The network activations $\hat{\varphi}(x) \approx \hat{A}^{\mathrm{T}} \varphi(x)$ are therefore a random rotation of the deterministic feature vector $\varphi(x)$. For the KPCA feature vector φ, $\hat{A}$ approximately computes an orthonormal change of coordinate of $\hat{\varphi}(x)$ to its PCA basis.

[1] We shall always assume that θ is the minimum-norm vector such that $f(x) = \langle \theta, \varphi(x) \rangle_H$.

For any function $f(x) = \langle \theta, \varphi(x) \rangle_H$ in $\mathcal{H}$, if the output layer weights are $\hat{\theta} = \hat{A}^{\mathrm{T}}\theta$, then the network output is

$$\hat{f}(x) = \langle \hat{A}^{\mathrm{T}}\theta, \hat{\varphi}(x) \rangle = \langle \theta, \hat{A}\,\hat{\varphi}(x) \rangle_H \approx f(x).$$

This means that the final layer coefficients $\hat{\theta}$ can cancel the random rotation $\hat{A}$ introduced by W, so that the random network output $\hat{f}(x)$ converges when the width d_1 increases to a fixed function in $\mathcal{H}$. This propagation of rotations across layers is key to understanding the weight dependencies in deep networks. We now make the above arguments more rigorous and prove that $\hat{\varphi}$ and $\hat{f}$ respectively converge to φ and f, for an appropriate choice of $\hat{A}$.

We write $\mathcal{O}(d_1)$ the set of linear operators A from $\mathbb{R}^{d_1}$ to $H = \ell^2(\mathbb{N})$ which satisfy $A^{\mathrm{T}}A = \mathrm{Id}_{d_1}$. Each $A \in \mathcal{O}(d_1)$ computes an isometric embedding of $\mathbb{R}^{d_1}$ into H, while A^{T} is an orthogonal projection onto a d_1-dimensional subspace of H which can be identified with $\mathbb{R}^{d_1}$. The alignment $\hat{A}$ of $\hat{\varphi}$ to φ is defined as the minimizer of the mean squared error:

$$\hat{A} = \operatorname*{arg\,min}_{A \in \mathcal{O}(d_1)} \mathbb{E}_x \Big[\| A\,\hat{\varphi}(x) - \varphi(x) \|_H^2 \Big]. \tag{7.3}$$

This optimization problem, known as the (orthogonal) Procrustes problem (Hurley and Cattell, 1962; Schönemann, 1966), admits a closed-form solution, computed from a singular value decomposition of the (uncentered) cross-covariance matrix between φ and $\hat{\varphi}$:

$$\hat{A} = UV^{\mathrm{T}} \quad \text{with} \quad \mathbb{E}_x \Big[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}} \Big] = USV^{\mathrm{T}}. \tag{7.4}$$

The mean squared error (7.3) of the optimal $\hat{A}$ (7.4) is then

$$\mathbb{E}_x \Big[\| \hat{A}\,\hat{\varphi}(x) - \varphi(x) \|_H^2 \Big] = \operatorname{tr} \mathbb{E}_x \Big[\hat{\varphi}(x)\,\hat{\varphi}(x)^{\mathrm{T}} \Big] + \operatorname{tr} \mathbb{E}_x \Big[\varphi(x)\,\varphi(x)^{\mathrm{T}} \Big] - 2 \Big\| \mathbb{E}_x \Big[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}} \Big] \Big\|_1, \tag{7.5}$$

where $\|\cdot\|_1$ is the nuclear (or trace) norm, that is, the sum of the singular values. Equation (7.5) defines a distance between the representations $\hat{\varphi}$ and φ which is related to various similarity measures used in the literature.[2]

The alignment rotation (7.3,7.4) was used by Haxby et al. (2011) to align fMRI response patterns of human visual cortex from different individuals, and by Smith et al. (2017) to align word embeddings from different languages. Alignment between network weights has also been considered in previous works, but it was restricted to permutation matrices (Entezari et al., 2022; Benzing et al., 2022; Ainsworth et al., 2022). Permutations have the advantage of commuting with pointwise non-linearities, and can therefore be introduced while exactly preserving the network output function. However, they are not sufficiently rich to capture the variability of random features. It is shown in Entezari et al. (2022) that the error after permutation alignment converges to zero with the number of random features d_1 at a polynomial rate which is cursed by the dimension d_0 of x. On the contrary, the following theorem proves that the error after rotation alignment has a convergence rate which is independent of the dimension d_0.

[2] By normalizing the variance of φ and $\hat{\varphi}$, eq. (7.5) can be turned into a similarity measure $\|\mathbb{E}_x[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}}]\|_1 / \sqrt{\mathbb{E}_x[\|\varphi(x)\|^2]\mathbb{E}_x[\|\hat{\varphi}(x)\|^2]}$. It is related to the kernel alignment used by Cristianini et al. (2001); Cortes et al. (2012); Kornblith et al. (2019), although the latter is based on the Frobenius norm of the cross-covariance matrix $\mathbb{E}_x[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}}]$ rather than the nuclear norm. Both similarity measures are invariant to rotations of either φ or $\hat{\varphi}$ and therefore only depend on the kernels k and $\hat{k}$, but the nuclear norm has a geometrical interpretation in terms of an explicit alignment rotation (7.4). Further, Appendix F.1 shows that the formulation (7.5) has connections to optimal transport through the Bures-Wasserstein distance (Bhatia et al., 2019). Canonical correlation analysis also provides an alignment, although not in the form of a rotation. It is based on a singular value decomposition of the cross-correlation matrix $\mathbb{E}_x[\varphi(x)\,\varphi(x)^{\mathrm{T}}]^{-1/2}\,\mathbb{E}_x[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}}]\,\mathbb{E}_x[\hat{\varphi}(x)\,\hat{\varphi}(x)^{\mathrm{T}}]^{-1/2}$ rather than the cross-covariance, and is thus sensitive to noise in the estimation of the covariance matrices (Raghu et al., 2017; Morcos et al., 2018). Equivalently, it corresponds to replacing φ and $\hat{\varphi}$ with their whitened counterparts $\mathbb{E}_x[\varphi(x)\,\varphi(x)^{\mathrm{T}}]^{-1/2}\,\varphi$ and $\mathbb{E}_x[\hat{\varphi}(x)\,\hat{\varphi}(x)^{\mathrm{T}}]^{-1/2}\,\hat{\varphi}$ in eqs. (7.3) to (7.5).

Theorem 7.1. *Assume that $\mathbb{E}_x[\|x\|^2] < +\infty$, ρ is Lipschitz, and π has finite fourth order moments. Then there exists a constant $c > 0$ which does not depend on d_0 nor d_1 such that*

$$\mathbb{E}_{W,x,x'}\left[|\hat{k}(x,x') - k(x,x')|^2\right] \leq c\, d_1^{-1},$$

where x' is an i.i.d. copy of x. Suppose that the sorted eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m \geq \cdots$ of $\mathbb{E}_x[\varphi(x)\,\varphi(x)^T]$ satisfy $\lambda_m = O(m^{-\alpha})$ with $\alpha > 1$. Then the alignment $\hat{A}$ defined in (7.3) satisfies

$$\mathbb{E}_{W,x}\left[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2\right] \leq c\, d_1^{-\eta} \quad \text{with} \quad \eta = \frac{\alpha - 1}{2(2\alpha - 1)} > 0.$$

Finally, for any $f(x) = \langle \theta, \varphi(x) \rangle_H$ in $\mathcal{H}$, if $\hat{\theta} = \hat{A}^{\mathrm{T}}\theta$ then

$$\mathbb{E}_{W,x}\left[|\hat{f}(x) - f(x)|^2\right] \leq c\, \|f\|_{\mathcal{H}}^2\, d_1^{-\eta}.$$

The proof is given in Appendix F.1. The convergence of the empirical kernel $\hat{k}$ to the asymptotic kernel k is a direct application of the law of large numbers. The mean-square distance (7.5) between $\hat{A}\,\hat{\varphi}$ and φ is then rewritten as the Bures-Wasserstein distance (Bhatia et al., 2019) between the kernel integral operators associated to $\hat{k}$ and k. It is controlled by their mean-square distance via an entropic regularization of the underlying optimal transport problem (see, e.g., Peyré and Cuturi, 2019). The convergence rate is then obtained by exploiting the eigenvalue decay of the kernel integral operator.

Theorem 7.1 proves that there exists a rotation $\hat{A}$ which nearly aligns the hidden layer of a random feature network with any feature vector of the asymptotic kernel, with an error which converges to zero. The network output converges if that same rotation is applied on the last layer weights. We will use this result in the next section to define deep rainbow networks, but we note that it can be of independent interest in the analysis of random feature representations. The theorem assumes a power-law decay of the covariance spectrum of the feature vector φ (which is independent of the choice of φ satisfying eq. (7.2)). Because $\sum_{m=1}^{\infty} \lambda_m = \mathbb{E}_x[\|\varphi(x)\|^2] < +\infty$ (as shown in the proof), a standard result implies that $\lambda_m = o(m^{-1})$, so the assumption $\alpha > 1$ is not too restrictive. The constant c is explicit and depends polynomially on the constants involved in the hypotheses (except for the exponent α). The convergence rate $\eta = \frac{\alpha-1}{2(2\alpha-1)}$ is an increasing function of the power-law exponent α. It vanishes in the critical regime when $\alpha \to 1$, and increases to $\frac{1}{4}$ when $\alpha \to \infty$. This bound might be pessimistic in practice, as a heuristic argument suggests a rate of $\frac{1}{2}$ when $\alpha \to \infty$ based on the rate 1 on the kernels. A comparison with convergence rates of random features KPCA (Sriperumbudur and Sterge, 2022) indeed suggests it might be possible to improve the convergence rate to $\frac{\alpha-1}{2\alpha-1}$. Although we give results in expectation for the sake of simplicity, bounds in probability can be obtained using Bernstein concentration bounds for operators (Tropp, 2012; Minsker, 2017) in the spirit of Rudi et al. (2013); Bach (2017b).

7.2.2 Deep rainbow networks

The previous section showed that the hidden layer of a random feature network converges to an infinite-dimensional feature vector, up to a rotation defined by the alignment $\hat{A}$. This section defines deep fully-connected rainbow networks by cascading conditional random features, whose kernels also converge in the infinite-width limit. It provides a model of the joint probability distribution of weights of trained networks, whose layer dependencies are captured by alignment rotation matrices.

We consider a deep fully-connected neural network with J hidden layers, which iteratively transforms the input data $x \in \mathbb{R}^{d_0}$ with weight matrices W_j of size $d_j \times d_{j-1}$ and a pointwise non-linearity ρ, to compute each activation layer of depth j:

$$\hat{\phi}_j(x) = \rho W_j \cdots \rho W_1\, x.$$

ρ includes a division by $\sqrt{d_j}$, which we do not write explicitly to simplify notations. After J non-linearities, the last layer outputs

$$\hat{f}(x) = \langle \hat{\theta}, \hat{\phi}_J(x) \rangle.$$

Infinite-width rainbow networks. A rainbow model defines each W_j conditionally on the previous $(W_\ell)_{\ell<j}$ as a random feature matrix. The distribution of random features at layer j is rotated to account for the random rotation introduced by $\hat{\phi}_{j-1}$. We first introduce infinite-width rainbow networks which define the asymptotic feature vectors used to compute these rotations.

Definition 7.1. *An infinite-width rainbow network has activation layers defined in a separable Hilbert space H_j for any $j \leq J$ by*

$$\phi_j(x) = \varphi_j(\varphi_{j-1}(\dots \varphi_1(x) \dots)) \in H_j \quad \text{for} \quad x \in H_0 = \mathbb{R}^{d_0},$$

where each $\varphi_j \colon H_{j-1} \to H_j$ is defined from a probability distribution π_j on H_{j-1} by

$$\langle \varphi_j(z), \varphi_j(z') \rangle_{H_j} = \mathbb{E}_{w \sim \pi_j} \Big[\rho(\langle z, w \rangle_{H_{j-1}}) \, \rho(\langle z', w \rangle_{H_{j-1}}) \Big] \quad \text{for} \quad z, z' \in H_{j-1}. \tag{7.6}$$

It defines a rainbow kernel

$$k_j(x, x') = \langle \phi_j(x), \phi_j(x') \rangle_{H_j}.$$

For $\theta \in H_J$, the infinite-width rainbow network outputs

$$f(x) = \langle \theta, \phi_J(x) \rangle_{H_J} \in \mathcal{H}_J,$$

where $\mathcal{H}_J$ is the RKHS of the rainbow kernel k_J of the last layer. If all probability distributions π_j are Gaussian, then the rainbow network is said to be Gaussian.

Each activation layer $\phi_j(x) \in H_j$ of an infinite-width rainbow network has an infinite dimension and is deterministic. We shall see that the cascaded feature maps φ_j are infinite-width limits of ρW_j up to rotations. One can arbitrarily rotate a feature vector $\varphi_j(z)$ which satisfies (7.6), which also rotates the Hilbert space H_j and $\phi_j(x)$. If the distribution π_{j+1} at the next layer (or the weight vector θ if $j = J$) is similarly rotated, this operation preserves the dot products $\langle \phi_j(x), w \rangle_{H_j}$ for $w \sim \pi_{j+1}$. It therefore does not affect the asymptotic rainbow kernels at each depth j:

$$k_j(x, x') = \mathbb{E}_{w \sim \pi_j} \Big[\rho(\langle \phi_{j-1}(x), w \rangle_{H_{j-1}}) \, \rho(\langle \phi_{j-1}(x'), w \rangle_{H_{j-1}}) \Big], \tag{7.7}$$

as well as the rainbow network output $f(x)$. We shall fix these rotations by choosing KPCA feature vectors. This imposes that $H_j = \ell^2(\mathbb{N})$ and $\mathbb{E}_x[\phi_j(x) \, \phi_j(x)^{\mathrm{T}}]$ is diagonal with decreasing values along the diagonal. The random feature distributions π_j are thus defined with respect to the PCA basis of $\phi_j(x)$. Infinite-width rainbow networks are then uniquely determined by the distributions π_j and the last-layer weights θ.

The weight distributions π_j for $j \geq 2$ are defined in the infinite-dimensional space H_{j-1} and some care must be taken. We say that a distribution π on a Hilbert space H has bounded second-order moments if its (uncentered) covariance operator $\mathbb{E}_{w \sim \pi}[w w^{\mathrm{T}}]$ is bounded (for the operator norm). The expectation is to be understood in a weak sense: we assume that there exists a bounded operator C on H such that $z^{\mathrm{T}} C z' = \mathbb{E}_{w \sim \pi}[\langle z, w \rangle_H \langle z', w \rangle_H]$ for $z, z' \in H$. We further say that π has bounded fourth-order moments if for every trace-class operator T (that is, such that $\mathrm{tr}(T^{\mathrm{T}} T)^{1/2} < +\infty$), $\mathbb{E}_{w \sim \pi}[(w^{\mathrm{T}} T w)^2] < +\infty$. We will assume that the weight distributions π_j have bounded second- and fourth-order moments. Together with our assumptions that $\mathbb{E}_x[\|x\|^2] < +\infty$ and that ρ is Lipschitz, this verifies the existence of all the

infinite-dimensional objects we will use in the sequel. For the sake of brevity, we shall not mention these verifications in the main text and defer them to Appendix F.2. Finally, we note that we can generalize rainbow networks to cylindrical measures π_j, which define cylindrical random variables w (Vakhania et al., 1987, see also Riedle, 2011 or Gawarecki and Mandrekar, 2011, Section 2.1.1). Such cylindrical random variables w are linear maps such that $w(z)$ is a real random variable for every $z \in H_{j-1}$. $w(z)$ cannot necessarily be written $\langle z, w \rangle$ with a random $w \in H_{j-1}$. We still write $\langle z, w \rangle$ by abuse of notation, with the understanding that it refers to $w(z)$. For example, we will see that finite-width networks at initialization converge to infinite-width rainbow networks with $\pi_j = \mathcal{N}(0, \mathrm{Id})$, which is a cylindrical measure but not a measure when H_{j-1} is infinite-dimensional.

Dimensionality reduction. Empirical observations of trained deep networks show that they have approximately low-rank weight matrices (Martin and Mahoney, 2021; Thamm et al., 2022). They compute a dimensionality reduction of their input, which is characterized by the singular values of the layer weight W_j, or equivalently the eigenvalues of the empirical weight covariance $d_j^{-1} W_j^{\mathrm{T}} W_j$. For rainbow networks, the uncentered covariances $C_j = \mathbb{E}_{w \sim \pi_j}[w w^{\mathrm{T}}]$ of the weight distributions π_j therefore capture the linear dimensionality reductions of the network. If $C_j^{1/2}$ is the symmetric square root of C_j, we can rewrite (7.6) with a change of variable as

$$\varphi_j(z) = \tilde{\varphi}_j\Big(C_j^{1/2} z\Big) \quad \text{with} \quad \langle \tilde{\varphi}_j(z), \tilde{\varphi}_j(z') \rangle_{H_j} = \mathbb{E}_{w \sim \tilde{\pi}_j}\Big[\rho(\langle z, w \rangle)\, \rho(\langle z', w \rangle)\Big],$$

where $\tilde{\pi}_j$ has an identity covariance. Rainbow network activations can thus be written:

$$\phi_j(x) = \tilde{\varphi}_j\, C_j^{1/2} \cdots \tilde{\varphi}_1\, C_1^{1/2}\, x. \tag{7.8}$$

Each square root $C_j^{1/2}$ performs a linear dimensionality reduction of its input, while the white random feature maps $\tilde{\varphi}_j$ compute high-dimensional non-linear embeddings. Such linear dimensionality reductions in-between kernel feature maps had been previously considered in previous works (Cho and Saul, 2009; Mairal, 2016; Bietti, 2019).

Gaussian rainbow networks. The distributions π_j are entirely specified by their covariance C_j for Gaussian rainbow networks, where we then have

$$\pi_j = \mathcal{N}(0, C_j).$$

When the covariance C_j is not trace-class, π_j is a cylindrical measure as explained above. If ρ is a homogeneous non-linearity such as ReLU, on can derive (Cho and Saul, 2009) from (7.7) that Gaussian rainbow kernels can be written from a homogeneous dot-product:

$$k_j(x, x') = \|z_j(x)\|\, \|z_j(x')\|\, \kappa\left(\frac{\langle z_j(x), z_j(x') \rangle}{\|z_j(x)\|\, \|z_j(x')\|}\right) \quad \text{with} \quad z_j(x) = C_j^{1/2} \phi_{j-1}(x), \tag{7.9}$$

where κ is a scalar function which depends on the non-linearity ρ. The Gaussian rainbow kernels k_j and the rainbow RKHS $\mathcal{H}_J$ only depend on the covariances $(C_j)_{j \leq J}$. If $C_j = \mathrm{Id}$ for each j, then k_j remains a dot-product kernel because $\langle z_j(x), z_j(x') \rangle = \langle \phi_{j-1}(x), \phi_{j-1}(x') \rangle = k_{j-1}(x, x')$. If the norms $\|z_j(x)\|$ concentrate, we then obtain $k_j(x, x') = \kappa(\dots \kappa(\langle x, x' \rangle) \dots)$ (Daniely et al., 2016). Depth is then useless, as k_j has the same expressivity as k_1 (Bietti and Bach, 2021). When $C_j \neq \mathrm{Id}$, Gaussian rainbow kernels k_j cannot be written as a cascade of elementary kernels, but their square roots ϕ_j are a cascade of kernel feature maps $\varphi_\ell = \tilde{\varphi}_\ell\, C_\ell^{1/2}$ for $\ell \leq j$. The white random feature maps $\tilde{\varphi}_j$ have simple expressions as they arise from the homogeneous dot-product kernel:

$$\langle \tilde{\varphi}_j(z), \tilde{\varphi}_j(z') \rangle_{H_j} = \|z\|\, \|z'\|\, \kappa\left(\frac{\langle z, z' \rangle}{\|z\|\, \|z'\|}\right).$$

This dot-product kernel implies that $\tilde{\varphi}_j$ is equivariant to rotations, and hence symmetry properties on the network ϕ_j as we will see in Section 7.2.3.

Finite-width rainbow networks. We now go back to the general case of arbitrary weight distributions π_j and introduce finite-width rainbow networks, which are random approximations of infinite-width rainbow networks. Each weight matrix W_j is iteratively defined conditionally on the previous weight matrices $(W_\ell)_{\ell<j}$. Its conditional probability distribution is defined in order to preserve the key induction property of the rainbow convergence of the activations $\hat{\phi}_j$. Informally, it states that $\hat{A}_j\,\hat{\phi}_j \approx \phi_j$ where $\hat{A}_j \colon \mathbb{R}^{d_j} \to H_j$ is an alignment rotation. Finite-width rainbow networks impose sufficient conditions to obtain this convergence at all layers, as we will show below.

The first layer W_1 is defined as in Section 7.2.1. Suppose that $W_1, \dots, W_{j-1}$ have been defined. By induction, there exists an alignment rotation $\hat{A}_{j-1} \colon \mathbb{R}^{d_{j-1}} \to H_{j-1}$, defined by

$$\hat{A}_{j-1} = \operatorname*{arg\,min}_{\hat{A} \in \mathcal{O}(d_{j-1})} \mathbb{E}_x \Big[\|\hat{A}\,\hat{\phi}_{j-1}(x) - \phi_{j-1}(x)\|^2_{H_{j-1}} \Big], \tag{7.10}$$

such that $\hat{A}_{j-1}\,\hat{\phi}_{j-1} \approx \phi_{j-1}$. We wish to define W_j so that $\hat{A}_j\,\hat{\phi}_j \approx \phi_j$. This can be achieved with a random feature approximation of φ_j composed with the alignment $\hat{A}_{j-1}$. Consider a (semi-infinite) random matrix W'_j of d_j i.i.d. rows in H_{j-1} distributed according to π_j:

$$W'_j = (w'_{ji})_{i \le d_j} \quad \text{with i.i.d.} \quad w'_{ji} \sim \pi_j.$$

We then have $\hat{A}_j\,\rho W'_j \approx \varphi_j$ for a suitably defined $\hat{A}_j$, as in Section 7.2.1. Combining the two approximations, we obtain

$$\hat{A}_j\,\rho W'_j\,\hat{A}_{j-1}\,\hat{\phi}_{j-1} \approx \varphi_j\,\phi_{j-1} = \phi_j.$$

We thus define the weight at layer j with the aligned random features

$$W_j = W'_j\,\hat{A}_{j-1}.$$

It is a random weight matrix of size $d_j \times d_{j-1}$, with rotated rows $\hat{A}^{\mathrm{T}}_{j-1} w'_{ji}$ that are independent and identically distributed when conditioned on the previous layers $(W_\ell)_{\ell<j}$. This inverse rotation of random weights cancels the rotation introduced by the random features at the previous layer, and implies a convergence of the random features cascade as we will prove below. This qualitative derivation motivates the following definition of finite-width rainbow networks.

Definition 7.2. *A finite-width rainbow network approximation of an infinite-width rainbow network with weight distributions $(\pi_j)_{j\le J}$ is defined for each $j \le J$ by a random weight matrix W_j of size $d_j \times d_{j-1}$ which satisfies*

$$W_j = (\hat{A}^T_{j-1} w'_{ji})_{i \le d_j} \quad \textit{with i.i.d.} \quad w'_{ji} \sim \pi_j, \tag{7.11}$$

where $\hat{A}_{j-1}$ is the rotation defined in (7.10). The last layer weight vector is $\hat{\theta} = \hat{A}^{\mathrm{T}}_J \theta$ where θ is the last layer weight of the infinite-width rainbow network.

The random weights W_j of a finite rainbow networks are defined as rotations and finite-dimensional projections of the d_j infinite-dimensional random vectors w'_{ji}, which are independent. The dependence on the previous layers $(W_\ell)_{\ell<j}$ is captured by the rotation $\hat{A}_{j-1}$. The rows of W_j are thus not independent, but they are independent when conditioned on $(W_\ell)_{\ell<j}$.

The rotation and projection of the random weights (7.11) implies a similar rotation and projection on the moments of W_j conditionally on $(W_\ell)_{\ell<j}$. In particular, the conditional covariance of W_j is thus

$$\hat{C}_j = \hat{A}_{j-1}^{\mathrm{T}} C_j \hat{A}_{j-1}. \tag{7.12}$$

W_j can then be factorized as the product of a white random feature matrix $\tilde{W}_j$ with the covariance square root:

$$W_j = \tilde{W}_j \, \hat{C}_j^{1/2} \quad \text{with i.i.d.} \quad \tilde{w}_{ji} \text{ conditionally on } (W_\ell)_{\ell<j}.$$

Note that the distribution of the white random features $\tilde{w}_{ji}$ depends in general on $\hat{A}_{j-1}$. However, for Gaussian rainbow networks with $\pi_j = \mathcal{N}(0, C_j)$, this dependence is limited to the covariance $\hat{C}_j$ and $\tilde{W}_j = G_j$ is a Gaussian white matrix with i.i.d. normal entries that are independent of the previous layer weights $(W_\ell)_{\ell<j}$:

$$W_j = G_j \, \hat{C}_j^{1/2} \quad \text{with i.i.d.} \quad G_{jik} \sim \mathcal{N}(0,1) \; . \tag{7.13}$$

Finite-width Gaussian rainbow networks are approximation models of deep networks that have been trained end-to-end by SGD on a supervised task. We will explain in Section 7.3 how each covariance C_j of the rainbow model can be estimated from the weights of one or several trained networks. The precision of a Gaussian rainbow model is evaluated by sampling new weights according to (7.13) and verifying that the resulting rainbow network has a similar performance as the original trained networks.

Convergence to infinite-width networks. The heuristic derivation used to motivate Definition 7.2 suggests that the weights rotation (7.11) guarantees the convergence of finite-width rainbow networks towards their infinite-width counterpart. This is proved by the next theorem, which builds on Theorem 7.1.

Theorem 7.2. *Assume that $\mathbb{E}_x[\|x\|^2] < +\infty$ and ρ is Lipschitz. Let $(\phi_j)_{j\leq J}$ be the activation layer of an infinite-width rainbow network with distributions $(\pi_j)_{j\leq J}$ with bounded second- and fourth-order moments, and an output $f(x)$. Let $(\hat{\phi}_j)_{j\leq J}$ be the activation layers of sizes $(d_j)_{j\leq J}$ of a finite-width rainbow network approximation, with an output $\hat{f}(x)$. Let $k_j(x,x') = \langle \phi_j(x), \phi_j(x') \rangle$ and $\hat{k}_j(x,x') = \langle \hat{\phi}_j(x), \hat{\phi}_j(x') \rangle$. Suppose that the sorted eigenvalues of $\mathbb{E}_x[\phi_j(x)\,\phi_j(x)^{\mathrm{T}}]$ satisfy $\lambda_{j,m} = O(m^{-\alpha_j})$ with $\alpha_j > 1$. Then there exists $c > 0$ which does not depend upon $(d_j)_{j\leq J}$ such that*

$$\begin{aligned}
\mathbb{E}_{W_1,\dots,W_j,x,x'}\Big[|\hat{k}_j(x,x') - k_j(x,x')|^2\Big] &\leq c\left(\varepsilon_{j-1} + d_j^{-1/2}\right)^2 \\
\mathbb{E}_{W_1,\dots,W_j,x}\Big[\|\hat{A}_j\,\hat{\phi}_j(x) - \phi_j(x)\|_{H_j}^2\Big] &\leq c\,\varepsilon_j^2 \\
\mathbb{E}_{W_1,\dots,W_J,x}\Big[|\hat{f}(x) - f(x)|^2\Big] &\leq c\,\|f\|_{\mathcal{H}_J}^2\,\varepsilon_J^2,
\end{aligned}$$

where

$$\varepsilon_j = \sum_{\ell=1}^{j} d_\ell^{-\eta_\ell/2} \quad \textit{with} \quad \eta_\ell = \frac{\alpha_\ell - 1}{2(2\alpha_\ell - 1)} > 0.$$

The proof is given in Appendix F.2. It applies iteratively Theorem 7.1 at each layer. As in Theorem 7.1, the constant c is explicit and depends polynomially on the constants involved in the hypotheses. For Gaussian weight distributions $\pi_j = \mathcal{N}(0, C_j)$, the theorem only requires that $\|C_j\|_\infty$ is finite for each $j \leq J$, where $\|\cdot\|_\infty$ is the operator norm (i.e., the largest singular value).

This theorem proves that at each layer, a finite-width rainbow network has an empirical kernel $\hat{k}_j$ which converges in mean-square to the deterministic kernel k_j of the infinite-width network, when all widths d_ℓ grow to infinity. Similarly, after alignment, each activation layer $\hat{\phi}_j$ also converges to the activation layer ϕ_j of the infinite-width network. Finally, the finite-width rainbow output $\hat{f}$ converges to a function f in the RKHS $\mathcal{H}_J$ of the infinite-width network. This demonstrates that all finite-width rainbow networks implement the same deterministic function when they are wide enough. Note that any relative scaling between the layer widths is allowed, as the error decomposes as a sum over layer contributions: each layer converges independently. In particular, this includes the proportional case when the widths are defined as $d_j = s\, d_j^0$ and the scaling factor s grows to infinity.

The asymptotic existence of rotations between any two trained networks has implications for the geometry of the loss landscape: if the weight distributions π_j are unimodal, which is the case for Gaussian distributions, alignment rotations can be used to build continuous paths in parameter space between the two rainbow network weights without encountering loss barriers (Freeman and Bruna, 2017; Draxler et al., 2018; Garipov et al., 2018). This could not be done with permutations (Entezari et al., 2022; Benzing et al., 2022; Ainsworth et al., 2022), which are discrete symmetries. It proves that under the rainbow assumptions, the loss landscape of wide-enough networks has a single connected basin, as opposed to many isolated ones.

Theorem 7.2 is a law-of-large-numbers result, which is different but complementary to the central-limit neural network Gaussian process convergence of Neal (1996); Williams (1996); Lee et al. (2018); Matthews et al. (2018). These works state that at initialization, random finite-dimensional projections of the activations $\hat{\phi}_j$ converge to a random Gaussian process described by a kernel. In contrast, we show in a wider setting that the activations $\hat{\phi}_j$ converge to a deterministic feature vector ϕ_j described by a more general kernel, up to a random rotation. Note that this requires no assumptions of Gaussianity on the weights or the activations. The convergence of the kernels is similar to the results of Daniely et al. (2016), but here generalized to non-compositional kernels obtained with arbitrary weight distributions π_j.

Theorem 7.2 can be considered as a multi-layer but static extension of the mean-field limit of Chizat and Bach (2018); Mei et al. (2018); Rotskoff and Vanden-Eijnden (2018); Sirignano and Spiliopoulos (2020). The limit is the infinite-width rainbow networks of Definition 7.1. It differs from other multi-layer extensions (Sirignano and Spiliopoulos, 2022; E and Wojtowytsch, 2020; Nguyen and Pham, 2020; Chen et al., 2022c; Yang and Hu, 2021) because Definition 7.2 includes the alignment rotations $\hat{A}_j$. We shall not model the optimization dynamics of rainbow networks when trained with SGD, but we will make several empirical observations in Section 7.3.

Finally, Theorem 7.2 shows that the two assumptions of Definition 7.2, namely that layer dependencies are reduced to alignment rotations and that neuron weights are conditionally i.i.d. at each layer, imply the convergence up to rotations of network activations at each layer. We will verify numerically this convergence in Section 7.3 for several network architectures on image classification tasks, corroborating the results of Raghu et al. (2017) and Kornblith et al. (2019). It does not mean that the assumptions of Definition 7.2 are valid, and verifying them is challenging in high-dimensions beyond the Gaussian case where the weight distributions π_j are not known. We however note that the rainbow assumptions are satisfied at initialization with $\pi_j = \mathcal{N}(0, \mathrm{Id})$, as eq. (7.12) implies that $\hat{C}_j = \mathrm{Id}$ and thus that the weight matrices $W_j = G_j$ are independent. Theorem 7.2 therefore applies at initialization. It is an open problem to show whether the existence of alignment rotations $\hat{A}_j$ is preserved during training by SGD, or whether dependencies between layer weights are indeed reduced to these rotations. Regarding (conditional) independence between neuron weights, Sirignano and Spiliopoulos (2020) show that in one-hidden-layer networks, neuron weights remain independent at non-zero but finite training times in the infinite-width limit. In contrast, a result of Rotskoff and Vanden-Eijnden (2018) suggests that this is no longer true at diverging training times, as SGD leads to an approximation of the target function f with a better rate than Monte-Carlo. Neuron weights at

a given layer remain however (conditionally) exchangeable due to the permutation equivariance of the initialization and SGD, and therefore have the same marginal distribution. Theorem 7.2 can be extended to dependent neuron weights w'_{ji}, e.g., with the more general assumption that their empirical distribution $d_j^{-1} \sum_{i=1}^{d_j} \delta_{w'_{ji}}$ converges weakly to π_j when the width d_j increases.

7.2.3 Symmetries and convolutional rainbow networks

The previous sections have defined fully-connected rainbow networks. In applications, prior information on the learning problem is often available. Practitioners then design more constrained architectures which implement inductive biases. Convolutional networks are important examples, which enforce two fundamental properties: equivariance to translations, achieved with weight sharing, and local receptive fields, achieved with small filter supports (LeCun et al., 1989a; LeCun and Bengio, 1995). We first explain how equivariance to general groups may be achieved in rainbow networks. We then generalize rainbow networks to convolutional architectures.

Equivariant rainbow networks. Prior information may be available in the form of a symmetry group under which the desired output is invariant. For instance, translating an image may not change its class. We now explain how to enforce symmetry properties in rainbow networks by imposing these symmetries on the weight distributions π_j rather than on the values of individual neuron weights w_{ji}. For Gaussian rainbow networks, we shall see that it is sufficient to impose that the desired symmetries commute with the weight covariances C_j.

Formally, let us consider G a subgroup of the orthogonal group $O(d_0)$, under whose action the target function $f^\star$ is invariant: $f^\star(gx) = f^\star(x)$ for all $g \in G$. Such invariance is generally achieved progressively through the network layers. In a convolutional network, translation invariance is built up by successive pooling operations. The output $f(x)$ is invariant but intermediate activations $\phi_j(x)$ are equivariant to the group action. Equivariance is more general than invariance. The activation map ϕ is equivariant if there is a representation σ of G such that $\phi(gx) = \sigma(g)\phi(x)$, where $\sigma(g)$ is an invertible linear operator such that $\sigma(gg') = \sigma(g)\sigma(g')$ for all $g, g' \in G$. An invariant function $f(x) = \langle \theta, \phi(x) \rangle$ is obtained from an equivariant activation map ϕ with a fixed point θ of the representation σ. Indeed, if $\sigma(g)\theta = \theta$ for all $g \in G$, then $f(gx) = f(x)$.

We say that σ is an orthogonal representation of G if $\sigma(g)$ is an orthogonal operator for all g. When σ is orthogonal, we say that ϕ is orthogonally equivariant. We also say that a distribution π is invariant under the action of σ if $\sigma(g)^{\mathrm{T}} w \sim \pi$ for all $g \in G$, where $w \sim \pi$. We say that a linear operator C commutes with σ if it commutes with $\sigma(g)$ for all $g \in G$. Finally, a kernel k is invariant to the action of G if $k(gx, gx') = k(x, x')$. The following theorem proves that rainbow kernels are invariant to a group action if each weight distribution π_j is invariant to the group representation on the activation layer ϕ_{j-1}, which inductively defines orthogonal representations σ_j at each layer.

Theorem 7.3. *Let G be a subgroup of the orthogonal group $O(d_0)$. If all weight distribution $(\pi_j)_{j \leq J}$ are invariant to the inductively defined orthogonal representation of G on their input activations, then activations $(\phi_j)_{j \leq J}$ are orthogonally equivariant to the action of G, and the rainbow kernels $(k_j)_{j \leq J}$ are invariant to the action of G. For Gaussian rainbow networks, this is equivalent to imposing that all weight covariances $(C_j)_{j \leq J}$ commute with the orthogonal representation of G on their input activations.*

The proof is in Appendix F.3. The result is proved by induction. If ϕ_j is orthogonally equivariant and π_{j+1} is invariant to its representation σ_j, then the next-layer activations are equivariant. Indeed, for $w \sim \pi_{j+1}$,

$$\rho\Big(\Big\langle \phi_j(gx), w \Big\rangle\Big) = \rho\Big(\Big\langle \sigma_j(g)\phi_j(x), w \Big\rangle\Big) = \rho\Big(\Big\langle \phi_j(x), \sigma_j(g)^{\mathrm{T}} w \Big\rangle\Big) \sim \rho\Big(\Big\langle \phi_j(x), w \Big\rangle\Big),$$

which defines an orthogonal representation σ_{j+1} on ϕ_{j+1}. Note that any distribution π_j which is invariant to an orthogonal representation σ_j necessarily has a covariance C_j which commutes with σ_j. The converse is true when π_j is Gaussian, which shows that Gaussian rainbow networks have a maximal number of symmetries among rainbow networks with weight covariances C_j.

Together with Theorem 7.2, Theorem 7.3 implies that finite-width rainbow networks can implement functions $\hat{f}$ which are approximately invariant, in the sense that the mean-square error $\mathbb{E}_{W_1,\dots,W_J,x}[|\hat{f}(gx) - \hat{f}(x)|^2]$ vanishes when the layer widths grow to infinity, with the same convergence rate as in Theorem 7.2. The activations $\hat{\phi}_j$ are approximately equivariant in a similar sense. This gives a relatively easy procedure to define neural networks having predefined symmetries. The usual approach is to impose that each weight matrix W_j is permutation-equivariant to the representation of the group action on each activation layer (Cohen and Welling, 2016; Kondor and Trivedi, 2018). This means that W_j is a group convolution operator and hence that the rows of W_j are invariant by this group action. This property requires weight-sharing or synchronization between weights of different neurons, which has been criticized as biologically implausible (Bartunov et al., 2018; Ott et al., 2020; Pogodin et al., 2021). On the contrary, rainbow networks implement symmetries by imposing that the neuron weights are independent samples of a distribution which is invariant under the group action. The synchronization is thus only at a global, statistical level. It also provides representations with the orthogonal group, which is much richer than the permutation group, and hence increases expressivity. It comes however at the cost of an approximate equivariance for finite layer widths.

Convolutional rainbow networks. Translation-equivariance could be achieved in a fully-connected architecture by imposing stationary weight distributions π_j. For Gaussian rainbow networks, this means that weight covariances C_j commute with translations, and are thus convolution operators. However, the weights then have a stationary Gaussian distribution and therefore cannot have a localized support. This localization has to be enforced with the architecture, by constraining the connectivity of the network. We generalize the rainbow construction to convolutional architectures, without necessarily imposing that the weights are Gaussian. It is achieved by a factorization of the weight layers, so that identical random features embeddings are computed for each patch of the input. As a result, all previous theoretical results carry over to the convolutional setting.

In convolutional networks, each W_j is a convolution operator which enforces both translation equivariance and locality. Typical architectures impose that convolutional filters have a predefined support with an output which may be subsampled. This architecture prior can be written as a factorization of the weight matrix:

$$W_j = L_j P_j,$$

where P_j is a prior convolutional operator which only acts along space and is replicated over channels (also known as depthwise convolution), while L_j is a learned pointwise (or 1×1) convolution which only acts along channels and is replicated over space. This factorization is always possible, and should not be confused with depthwise-separable convolutions (Sifre and Mallat, 2013; Chollet, 2017).

Let us consider a convolutional operator W_j having a spatial support of size s_j^2, with d_{j-1} input channels and d_j output channels. The prior operator P_j then extracts d_{j-1} patches of size $s_j \times s_j$ at each spatial location and reshapes them as a channel vector of size $d'_{j-1} = d_{j-1} s_j^2$. P_j is fixed during training and represents the architectural constraints imposed by the convolutional layer. The learned operator L_j is then a 1×1 convolutional operator, applied at each spatial location across d'_{j-1} input channels to compute d_j output channels. This factorization reshapes the convolution kernel of W_j of size $d_j \times d_{j-1} \times s_j \times s_j$ into a 1×1 convolution L_j with a kernel of size $d_j \times d'_{j-1} \times 1 \times 1$. L_j can then be thought as a fully-connected operator over channels that is applied at every spatial location.

The choice of the prior operator P_j directly influences the learned operator L_j and therefore the weight distributions π_j. P_j may thus be designed to achieve certain desired properties on π_j. For instance, the operator P_j may also specify predefined filters, such as wavelets in learned scattering networks introduced in Chapters 5 and 6. In a learned scattering network, P_j computes spatial convolutions and subsamplings, with q wavelet filters having different orientations and frequency selectivity. The learned convolution L_j then has $d'_{j-1} = d_{j-1}q$ input channels. This is further detailed in Appendix F.4, which explains that one can reduce the size of L_j by imposing that it commutes with P_j, which amounts to factorizing $W_j = P_j L_j$ instead.

The rainbow construction of Section 7.2.2 has a straightforward extension to the convolutional case, with a few adaptations. The activations layers $\hat{\phi}_{j-1}$ should be replaced with $P_j\hat{\phi}_{j-1}$ and W_j with L_j, where it is understood that it represents a fully-connected matrix acting along channels and replicated pointwise across space. Similarly, the weight covariances C_j and its square roots $C_j^{1/2}$ are 1×1 convolutional operators which act along the channels of $P_j\hat{\phi}_{j-1}$, or equivalently are applied over patches of $\hat{\phi}_{j-1}$. Finally, the alignments $\hat{A}_{j-1}$ are 1×1 convolutions which therefore commute with P_j as they act along different axes. One can thus still define $\hat{C}_j = \hat{A}_{j-1}^{\mathrm{T}} C_j \hat{A}_{j-1}$. Convolutional rainbow networks also satisfy Theorems 7.1 to 7.3 with appropriate modifications.

We note that the expression of the rainbow kernel is different for convolutional architectures. Equation (7.7) becomes

$$k_j(x,x') = \sum_u \mathbb{E}_{w\sim\pi_j}\Big[\rho(\langle P_j\phi_{j-1}(x)[u], w\rangle)\rho(\langle P_j\phi_{j-1}(x')[u], w\rangle)\Big],$$

where $P_j\phi_{j-1}(x)[u]$ is a patch of $\phi_{j-1}(x)$ centered at u and whose spatial size is determined by P_j. In the particular case where π_j is Gaussian with a covariance C_j, the dot-product kernel in eq. (7.9) becomes

$$k_j(x,x') = \sum_u \|z_u(x)\|\,\|z_u(x')\|\,\kappa\left(\frac{\langle z_u(x), z_u(x')\rangle}{\|z_u(x)\|\,\|z_u(x')\|}\right) \quad \text{with} \quad z_u(x) = C_j^{1/2} P_j\phi_{j-1}(x)[u],$$

The sum on the spatial location u averages the local dot-product kernel values and defines a translation-invariant kernel. Observe that it differs from the fully-connected rainbow kernel (7.9) with weight covariances $C'_j = P_j^{\mathrm{T}} C_j P_j$, which is a global dot-product kernel with a stationary covariance. Indeed, the corresponding fully-connected rainbow networks have filters with global spatial support, while convolutional rainbow networks have localized filters. The covariance structure of depthwise convolutional filters has been investigated by Trockman et al. (2023).

The architecture plays an important role by modifying the kernel and hence the RKHS $\mathcal{H}_J$ of the output (Daniely et al., 2016). Hierarchical convolutional kernels have been studied by Mairal et al. (2014); Anselmi et al. (2015); Bietti (2019). Bietti and Mairal (2019) have proved that functions in $\mathcal{H}_J$ are stable to the action of diffeomorphisms (Mallat, 2012) when P_j also include a local averaging before the patch extraction. However, the generalization properties of such kernels are not well understood, even when $C_j = \mathrm{Id}$. In that case, deep kernels with $J > 1$ hidden layers are not equivalent to shallow kernels with $J = 1$ (Bietti and Bach, 2021).

7.3 Numerical results

In this section, we validate the rainbow model on several network architectures trained on image classification tasks and make several observations on the properties of the learned weight covariances C_j. As our first main result, we partially validate the rainbow model by showing that network activations converge up to rotations when the layer widths increase (Section 7.3.1). We then show in Section 7.3.2 that the empirical weight covariances $\hat{C}_j$ converge up to rotations when the layer widths increase. Furthermore, the weight covariances are typically low-rank and

can be partially specified from the input activation covariances. Our second main result, in Section 7.3.3, is that the Gaussian rainbow model applies to scattering networks trained on the CIFAR-10 dataset. Generating new weights from the estimated covariances C_j leads to similar performance than SGD training when the network width is large enough. We further show that SGD only updates the weight covariance during training while preserving the white Gaussian initialization. It suggests a possible explanation for the Gaussian rainbow model, though the Gaussian assumption seems too strong to hold for more complex learning tasks for network widths used in practice.

7.3.1 Convergence of activations in the infinite-width limit

We show that trained networks with different initializations converge to the same function when their width increases. More precisely, we show the stronger property that at each layer, their activations converge after alignment to a fixed deterministic limit when the width increases. Trained networks thus share the convergence properties of rainbow networks (Theorem 7.2). Section 7.3.3 will further show that scattering networks trained on CIFAR-10 indeed approximate Gaussian rainbow networks. In this case, the limit function is thus in the Gaussian rainbow RKHS (Definition 7.1).

Architectures and tasks. In this chapter, we consider two architectures, learned scattering networks and ResNets (He et al., 2016), trained on two image classification datasets, CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015).

Scattering networks have fixed spatial filters, so that their learned weights only operate across channels. This structure reduces the learning problem to channel matrices and plays a major role in the (conditional) Gaussianity of the learned weights, as we will see. The networks have J hidden layers, with $J = 7$ on CIFAR-10 and $J = 10$ on ImageNet. Each layer can be written $W_j = L_j P_j$ where L_j is a learned 1×1 convolution, and P_j is a convolution with predefined complex wavelets. P_j convolves each of its d_{j-1} input channels with 5 different wavelet filters (1 low-frequency filter and 4 oriented high-frequency wavelets), thus generating $d'_{j-1} = 5d_{j-1}$ channels. We shall still denote L_j with W_j to keep the notations of Section 7.2.2. The non-linearity ρ is a complex modulus with skip-connection, followed by a standardization (as computed by a batch-normalization). This architecture is slightly adapted from Chapter 6 and is further detailed in Appendix F.4.

Our scattering network reaches an accuracy of 92% on the CIFAR-10 test set. As a comparison, ResNet-20 (He et al., 2016) achieves 91% accuracy, while most linear classification methods based on hierarchical convolutional kernels such as the scattering transform or the neural tangent kernel reach less than 83% accuracy (Mairal et al., 2014; Oyallon and Mallat, 2015; Li et al., 2019). On the ImageNet dataset (Russakovsky et al., 2015), learned scattering networks achieve 89% top-5 accuracy, which is also the performance of ResNet-18 with single-crop testing.

We have made minor adjustments to the ResNet architecture for ease of analysis such as removing bias parameters (at no cost in performance), as explained in Appendix F.4. It can still be written $W_j = L_j P_j$ where P_j is a patch extraction operator as explained in Section 7.2.3, and the non-linearity ρ is a ReLU.

Convergence of activations. We train several networks with a range of widths by simultaneously scaling the widths of all layers with a multiplicative factor s varying over a range of $2^6 = 64$. We show that their activations $\hat{\phi}_j$ converge after alignment to a fixed deterministic limit ϕ_j when the width increases. The feature map ϕ_j is approximated with the activations of a large network with $s = 2^3$.

We begin illustrating the behavior of activation spectra as a function of our width-scaling parameter s, for seven-hidden-layer trained scattering networks on CIFAR-10. In the left panel

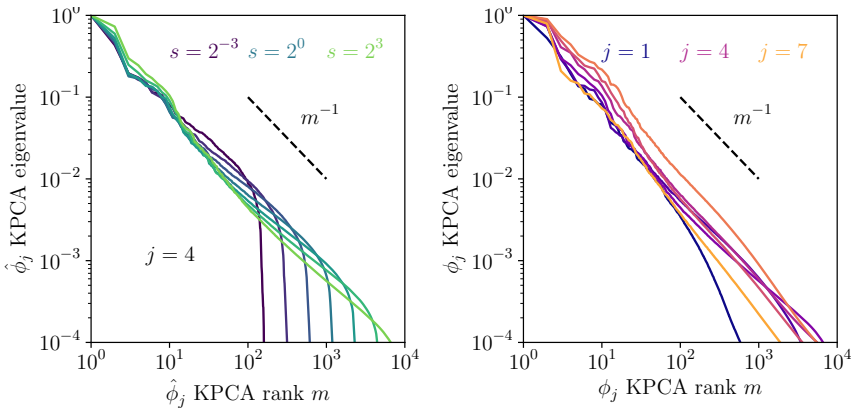

*FIGURE 7.2: Convergence of spectra of activations $\hat{\phi}_j$ of finite-width trained scattering networks towards the feature vector ϕ_j. The figure shows the covariance spectra of activations $\hat{\phi}_j$ for a given layer $j = 4$ and various width scaling s (*left*) and of the feature vector ϕ_j for the seven hidden layers $j \in \{1, \dots, 7\}$ (*right*). The covariance spectrum is a power law of index close to -1.*

of Figure 7.2, we show how activation spectra vary as a function of s for the layer $j = 4$ which has a behavior representative of all other layers. The spectra are obtained by doing a PCA of the activations $\hat{\phi}_j(x)$, which corresponds to a KPCA of the input x with respect to the empirical kernel $\hat{k}_j$. The $\hat{\phi}_j$ covariance spectra for networks of various widths overlap at lower KPCA ranks, suggesting well-estimated components, while the variance then decays rapidly at higher ranks. Wider networks thus estimate a larger number of principal components of the feature vector ϕ_j. For the first layer $j = 1$, this recovers the random feature KPCA results of Sriperumbudur and Sterge (2022), but this convergence is observed at all layers. The overall trend as a function of s illustrates the infinite-width convergence. We also note that, as the width increases, the activation spectrum becomes closer to a power-law distribution with a slope of -1. The right panel of the figure shows that this type of decay with KPCA rank m is observed at all layers of the infinite-width network $(\phi_j)_{j \leq J}$. The power-law spectral properties of random feature activations have been studied theoretically by Scetbon and Harchaoui (2021), and in connection with the scaling laws observed in large language models (Kaplan et al., 2020) by Maloney et al. (2022). Note that here we do not scale the dataset size nor training hyperparameters such as the learning rate or batch size with the network width, and a different experimental setup would likely influence the infinite-width limit (Yang et al., 2022; Hoffmann et al., 2022).

We now directly measure the convergence of activations by evaluating the mean-square distance after alignment $\mathbb{E}_x[\|\hat{A}_j \hat{\phi}_j(x) - \phi_j(x)\|^2]$. The left panel of Figure 7.3 shows that it does indeed decrease when the network width increases, for all layers j. Despite the theoretical convergence rate of Theorem 7.2 vanishing when the activation spectrum exponent α_j approaches 1, in practice we still observe convergence. Alignment rotations $\hat{A}_j$ are computed on the train set while the mean-square distance is computed on the test set, so this decrease is not a result of overfitting. It demonstrates that scattering networks $\hat{\phi}_j$ approximate the same deterministic network ϕ_j no matter their initialization or width when it is large enough. The right panel of the figure evaluates this same convergence on a ResNet-18 trained on ImageNet. The mean-square distance after alignment decreases for most layers when the width increases. We note that the rate of decrease slows down for the last few layers. For these layers, the relative error after alignment is of the order of unity, indicating that the convergence is not observed at the largest width considered here. The overall trend however suggests that further increasing the width would reduce the error after alignment. The observations that networks trained from different initializations have similar activations had already been made by Raghu et al. (2017). Kornblith et al. (2019) showed that similarity increases with width, but with a weaker similarity measure. Rainbow networks, which we will show can approximate scattering networks, explain the source

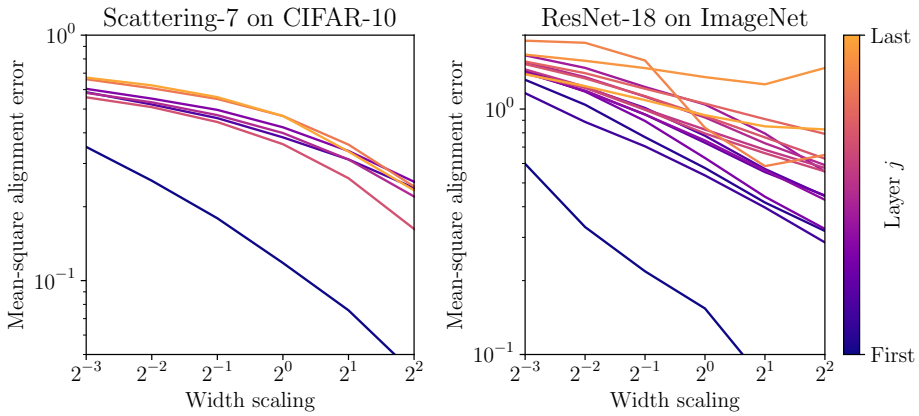

*FIGURE 7.3: Convergence of activations $\hat{\phi}_j$ of finite-width networks towards the corresponding feature vector ϕ_j, for scattering networks trained on CIFAR-10 (*left*) and ResNet trained on ImageNet (*right*). Both panels show the relative mean squared error $\mathbb{E}_x[\|\hat{A}_j\,\hat{\phi}_j(x) - \phi_j(x)\|^2]/\mathbb{E}_x[\|\phi_j(x)\|^2]$ between aligned activations $\hat{A}_j\,\hat{\phi}_j$ and the feature vector ϕ_j. The error decreases as a function of the width scaling s for all layers for the scattering network, and all but the last few layers for ResNet.*

of these observations as a consequence of the law of large numbers applied to the random weight matrices with conditionally i.i.d. rows.

7.3.2 Properties of learned weight covariances

We have established the convergence (up to rotations) of the activations $\hat{\phi}_j$ in the infinite-width limit. Under the rainbow model, the weight matrices W_j are random and thus cannot converge. However, they define estimates $\tilde{C}_j$ of the infinite-dimensional weight covariances C_j. We show that these estimates $\tilde{C}_j$ converge to the true covariances C_j when the width increases. We then demonstrate that the covariances C_j are effectively low-rank, and that their eigenspaces can be efficiently approximated by taking into account unsupervised information. The weight covariances are thus of low complexity, in the sense that they can be described with a number of parameters significantly smaller than their original size.

Estimation of the weight covariances. We estimate the weight covariances C_j from the learned weights of a deep network. This network has weight matrices W_j of size $d_j \times d_{j-1}$ that have been trained end-to-end by SGD. The natural empirical estimate of the weight covariance $\hat{C}_j$ of W_j is

$$\hat{C}_j \approx d_j^{-1}\, W_j^{\mathrm{T}} W_j. \tag{7.14}$$

It computes $\hat{C}_j$ from d_j samples, which are conditionally i.i.d. under the rainbow model hypothesis. Although the number d_j of samples is large, their dimension d_{j-1} is also large. For many architectures d_j/d_{j-1} remains nearly constant and we shall consider in this section that $d_j = s\, d_j^0$, so that when the scaling factor s grows to infinity d_j/d_{j-1} converges to a non-zero finite limit. This creates challenges in the estimation of $\hat{C}_j$, as we now explain. We will see that the weight variance is amplified during training. The learned covariance can thus be modeled $\hat{C}_j = \mathrm{Id} + \hat{C}'_j$, where the magnitude of $\hat{C}'_j$ keeps increasing during training. When the training time goes to infinity, the initialization Id becomes negligible with respect to $\hat{C}'_j$. However, at finite training time, only the eigenvectors of C'_j with sufficiently high eigenvalues have been learned consistently, and $\hat{C}'_j$ is thus effectively low-rank. $\hat{C}_j$ is then a spiked covariance matrix (Johnstone, 2001). A large statistical literature has addressed the estimation of spiked covariances when the number of parameters d_{j-1} and the number of observations d_j increases, with a constant ratio d_j/d_{j-1} (Baik et al., 2005; El Karoui, 2008a). Consistent estimators of the eigenvalues of $\hat{C}_j$ can

be computed, but not of its eigenvectors, unless we have other prior information such as sparsity of the covariance entries (El Karoui, 2008b) or its eigenvectors (Ma, 2013). In our setting, we shall see that prior information on eigenspaces of $\hat{C}_j$ is available from the eigenspaces of the input activation covariances. We use the empirical estimator (7.14) for simplicity, but it is not optimal. Minimax-optimal estimators are obtained by shrinking empirical eigenvalues (Donoho et al., 2018).

We would like to estimate the infinite-dimensional covariances C_j rather than finite-dimensional projections $\hat{C}_j$. Since $\hat{C}_j = \hat{A}_{j-1}^{\mathrm{T}} C_j \hat{A}_{j-1}$, an empirical estimate of C_j is given by

$$\tilde{C}_j = \hat{A}_{j-1} \hat{C}_j \hat{A}_{j-1}^{\mathrm{T}}. \tag{7.15}$$

To compute the alignment rotation $\hat{A}_{j-1}$ with eq. (7.4), we must estimate the infinite-width rainbow activations ϕ_{j-1}. As above, we approximate ϕ_{j-1} with the activations $\hat{\phi}_{j-1}$ of a finite but sufficiently large network, relying on the activation convergence demonstrated in the previous section. We then estimate C_j with eq. (7.15) and $\hat{C}_j \approx d_j^{-1} W_j^{\mathrm{T}} W_j$. We further reduce the estimation error of C_j by training several networks of size $(d_j)_{j \leq J}$, and by averaging the empirical estimators (7.15). Note that averaging directly the estimates (7.14) of $\hat{C}_j$ with different networks would not lead to an estimate of C_j, because the covariances $\hat{C}_j$ are represented in different bases which must be aligned. The final layer weights θ are also similarly computed with an empirical estimator from the trained weights $\hat{\theta}$.

Convergence of weight covariances. We now show numerically that the weight covariance estimates $\tilde{C}_j$ (7.15) converge to the true covariances C_j. This performs a partial validation of the rainbow assumptions of Definition 7.2, as it verifies the rotation of the second-order moments of π_j (7.12) but not higher-moments nor independence between neurons. Due to computational limitations, we perform this verification on three-hidden-layer scattering networks trained on CIFAR-10, for which we can scale both the number of networks N we can average over, and their width s. The main computational bottleneck here is the singular value decomposition of the cross-covariance matrix $\mathbb{E}_x[\phi_j(x)\, \hat{\phi}_j(x)^{\mathrm{T}}]$ to compute the alignment $\hat{A}_j$, which requires $O(Ns^3)$ time and $O(Ns^2)$ memory. These shallower networks reach a test accuracy of 84% at large width.

We begin by showing that empirical covariance matrices $\tilde{C}_j$ estimated from the weights of different networks share the same eigenspaces of large eigenvalues. To this end, we train N networks of the same finite width ($s = 1$) and compare the covariances $\tilde{C}_j$ estimated from these N networks as a function of N. As introduced above, the estimated covariances $\tilde{C}_j$ are well modeled with a spiked-covariance model. The upper-left panel of Figure 7.4 indeed shows that the covariance spectrum interpolates between an exponential decay at low ranks (indicated by the dashed line, corresponding to the "spikes" resulting from training, as will be shown in Section 7.3.3), and a Marchenko-Pastur tail at higher ranks (indicated by dotted lines, corresponding to the initialization with identity covariance). Note that we show the eigenvalues as a function of their rank rather than a spectral density in order to reveal the exponential decay of the spike positions with rank, which was missed in previous works (Martin and Mahoney, 2021; Thamm et al., 2022). The exponential regime is present even in the covariance estimated from a single network, indicating its stability across training runs, while the Marchenko-Pastur tail becomes flatter as more samples are used to estimate the empirical covariance. Here, the feature vector ϕ_j has been estimated with a scattering network of same width $s = 1$ for simplicity of illustration.

As shown in the lower-left panel, only the exponential regime contributes to the classification accuracy of the network: the neuron weights can be projected on the first principal components of $\tilde{C}_j$, which correspond to the learned spikes, without harming performance. The informative component of the weights is thus much lower-dimensional (≈ 30) than the network width (128),

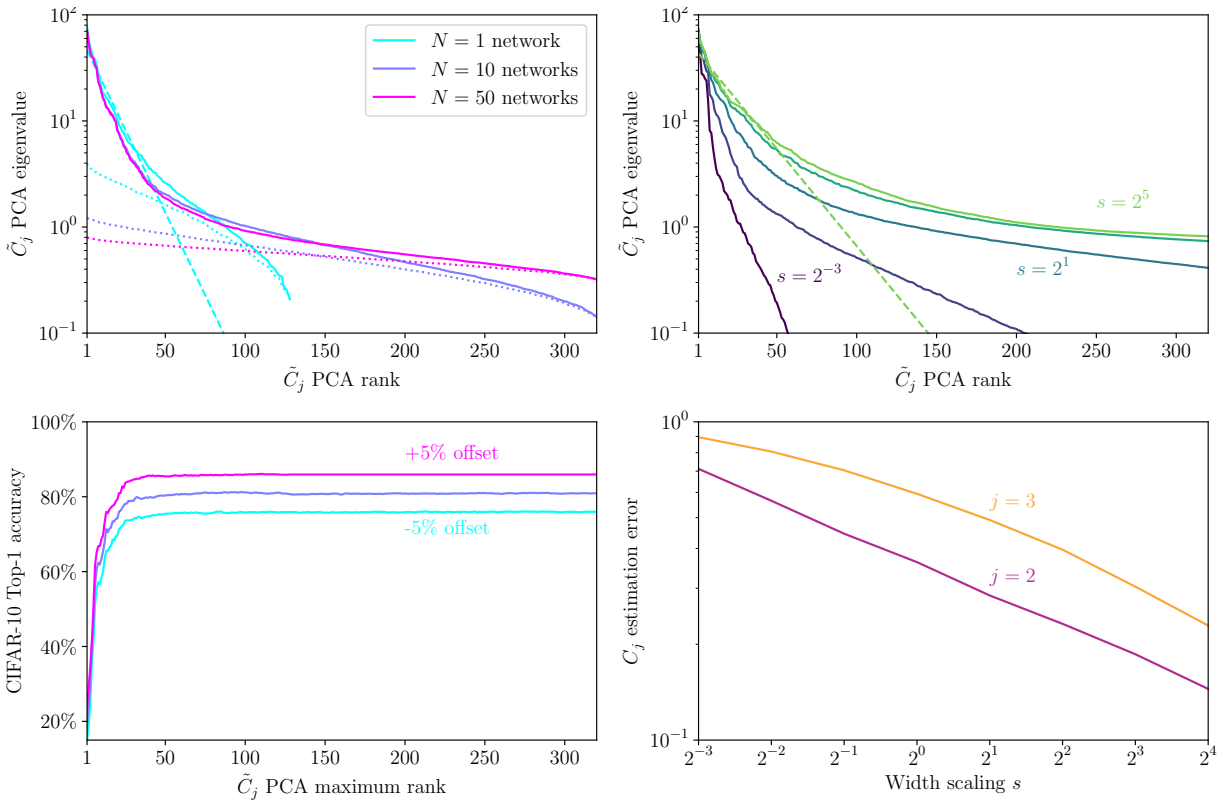

FIGURE 7.4: The weight covariance estimate $\tilde{C}_j$ converges towards the infinite-dimensional covariance C_j for a three-hidden-layer scattering network trained on CIFAR-10. The first three panels show the behavior of the layer $j = 2$. Upper left*: spectra of empirical weight covariances $\tilde{C}_j$ as a function of the network sample size N showing the transition from an exponential decay (fitted by the dashed line for $N = 1$) to the Marchenko-Pastur spectrum (fitted by the dotted lines).* Lower left*: test classification performance on CIFAR-10 of the trained networks as a function of the maximum rank of its weight covariance $\tilde{C}_j$. Most of the performance is captured with the first eigenvectors of $\tilde{C}_j$. The curves for different network sample sizes N when estimating $\tilde{C}_j$ overlap and are offset for visual purposes.* Upper right*: spectrum of empirical weight covariances $\tilde{C}_j$ as a function of the network width scaling s. The dashed line is a fit to an exponential decay at low rank.* Lower right*: relative distance between empirical and true covariances $\|\hat{C}_j - C_j\|_\infty / \|C_j\|_\infty$, as a function of the width scaling s.*

and this dimension appears to match the characteristic scale of the exponential decay of the covariance eigenvalues. The number N of trained networks used to compute $\tilde{C}_j$ has no appreciable effect on the approximation accuracy, which again shows that the empirical covariance matrices of all N networks share this common informative component. This presence of a low-dimensional informative weight component is in agreement with the observation that the Hessian of the loss at the end of training is dominated by a subset of its eigenvectors (LeCun et al., 1989b; Hassibi and Stork, 1992). These Hessian eigenvectors could indeed be related to the weight covariance eigenvectors. Similarly, the dichotomy in weight properties highlighted by our analysis could indicate why the eigenvalue distribution of the loss Hessian separates into two distinct regimes (Sagun et al., 2016, 2017; Papyan, 2019): the "bulk" (with small eigenvalues corresponding to uninformative flat directions of the loss landscape) is related to the Marchenko-Pastur tail of our weight covariance spectrum and the "top" (or spiked) components correspond to the exponential regime found at the lowest ranks of the covariance spectrum.

We now demonstrate that the weight covariances $\tilde{C}_j$ converge to an infinite-dimensional covariance operator C_j when the widths of the scattering networks increase. Here, the weight covariances $\tilde{C}_j$ are estimated from the weights of $N = 10$ networks with the same width scaling s, and we estimate C_j from the weights of $N = 10$ wide scattering networks with $s = 2^5$. We

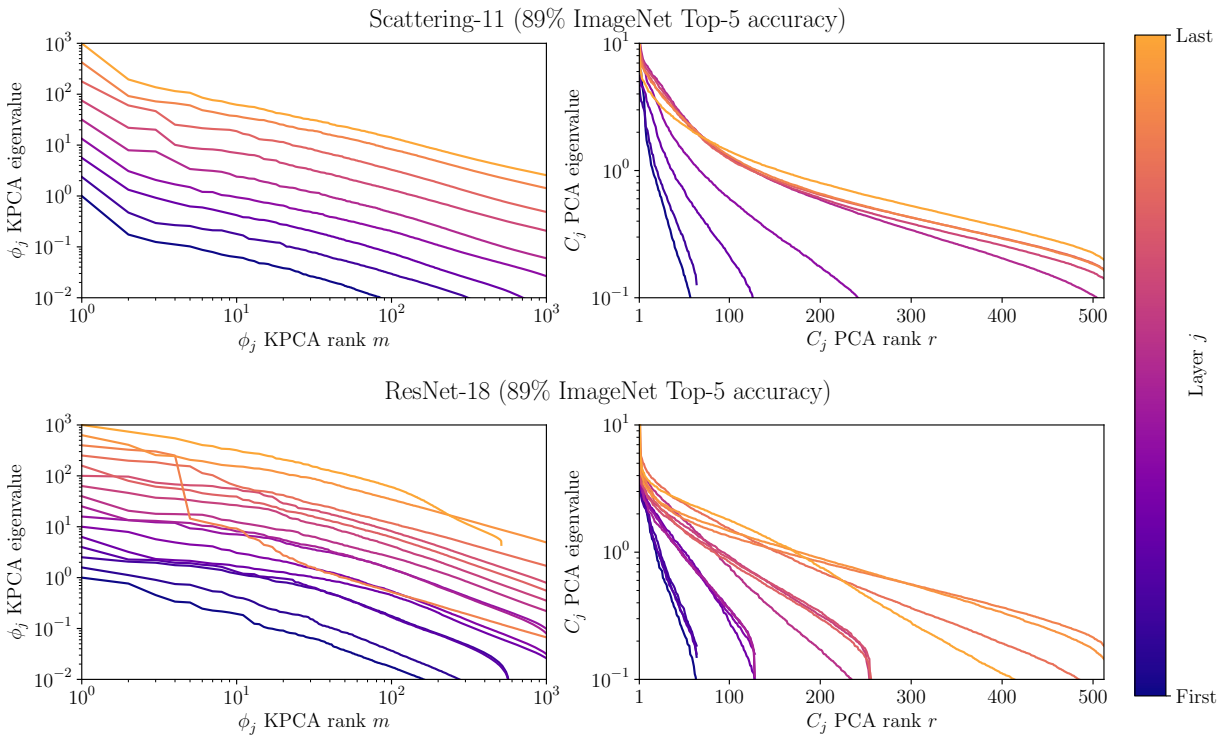

*Figure 7.5: Covariance spectra of activations and weights of an ten-hidden-layer scattering network (*top*) and ResNet-18 (*bottom*) trained on ImageNet. In both cases, activation spectra (*left*) mainly follow power-law distribution with index roughly -1. Weight spectra (*right*) show a transition from an exponential decay with a characteristic scale increasing with depth to the Marchenko-Pastur spectral distribution. These behaviors are captured by the rainbow model. For visual purposes, activation and weight spectra are offset by a factor depending on j. In addition, we do not show the first layer nor the 1×1 convolutional residual branches in ResNet as they have different layer properties.*

first illustrate this convergence on the spectrum of $\tilde{C}_j$ in the upper-right panel of Figure 7.2. The entire spectrum of $\tilde{C}_j$ converges to a limiting spectrum which contains both the informative exponential part resulting from training and the uninformative Marchenko-Pastur tail coming from the initialization. The characteristic scale of the exponential regime grows with network width but converges to a finite value as the width increases to infinity. We then confirm that the estimated covariances $\tilde{C}_j$ indeed converge to the covariance C_j when the width increases in the lower-right panel. The distance converges to zero as a power law of the width scaling. The first layer $j = 1$ has a different convergence behavior (not shown) as its input dimension does not increase with s.

In summary, in the context considered here, networks trained from different initializations share the same informative weight subspaces (after alignment) described by the weight covariances at each layer, and they converge to a deterministic limit when the width increases. The following paragraphs then demonstrate several properties of the weight covariances.

Dimensionality reduction in deep networks. We now consider deeper networks and show that they also learn low-rank covariances. Comparing the spectra of weights and activations reveals the alternation between dimensionality reduction with the colored weight covariances C_j and high-dimensional embeddings with the white random features which are captured in the rainbow model. We do so with two architectures: a ten-hidden-layer scattering network and a slightly modified ResNet-18 trained on ImageNet (specified in Appendix F.4), which both reach 89% top-5 test accuracy.

We show the spectra of covariances of activations ϕ_j in the left panels of Figure 7.5 and of

the weight covariances C_j in the right panels. For both networks, we recover the trend that activation spectra are close to power laws of slope -1 and the weight spectra show a transition from a learned exponential regime to a decay consistent with the Marchenko-Pastur expectation, which is almost absent for ResNet-18. Considering them in sequence, as a function of depth, the input activations are thus high-dimensional (due to the power-law of index close to -1) while the subsequent weights perform a dimensionality reduction using an exponential bottleneck with a characteristic scale much smaller than the width. Next, the dimensionality is re-expanded with the non-linearity, as the activations at the next layer again have a power-law covariance spectrum. Considering the weight spectra, we observe that the effective exponential scale increases with depth, from about 10 to 60 for both the scattering network and the ResNet. This increase of dimensionality with depth is expected: in convolutional architectures, the weight covariances C_j are only defined on small patches of activations ϕ_{j-1} because of the prior operator P_j. However, these patches correspond to a larger receptive field in the input image x as the depth j increases. The rank of the covariances is thus to be compared with the size of this receptive field. Deep convolutional networks thus implement a sequence of dimensionality contractions (with the learned weight covariances) and expansions (with the white random features and non-linearity). Without the expansion, the network would reduce the dimensionality of the data exponentially fast with depth, thus severely limiting its ability to process information on larger spatial scales (deeper layers), while without the contraction, its parameter count and learning sample complexity would increase exponentially fast with depth. This contraction/expansion strategy allows the network to maintain a balanced representation at each scale.

The successive increases and decreases in dimensionality due to the weights and non-linearity across deep network layers have been observed by Recanatesi et al. (2019) with a different dimensionality measure. The observation that weight matrices of trained networks are low-rank has been made in several works which exploited it for model compression (Denil et al., 2013; Denton et al., 2014; Yu et al., 2017), while the high-dimensional embedding property of random feature maps is well-known via the connection to their kernel (Rahimi and Recht, 2007; Scetbon and Harchaoui, 2021). The rainbow model integrates these two properties. In neuroscience, high-dimensional representations with power-law spectra have been measured in the mouse visual cortex by Stringer et al. (2019). Such representations in deep networks have been demonstrated to lead to increased predictive power of human fMRI cortical responses (Elmoznino and Bonner, 2022) and generalization in self-supervised learning (Agrawal et al., 2022).

Unsupervised approximations of weight covariances. The learning complexity of a rainbow network depends upon the number of parameters needed to specify the weight covariances $(C_j)_{j \leq J}$ to reach a given performance. After having shown that their informative subspace is of dimension significantly lower than the network width, we now show that this subspace can be efficiently approximated by taking into account unsupervised information.

We would like to define a representation of the weight covariances C_j which can be accurately approximated with a limited number of parameters. We chose to represent the infinite-width activations ϕ_j as KPCA feature vectors, whose uncentered covariances $\mathbb{E}_x[\phi_j(x)\,\phi_j(x)^{\mathrm{T}}]$ are diagonal. In that case, the weight covariances C_j for $j > 1$ are operators defined on $H_{j-1} = \ell^2(\mathbb{N})$. It amounts to representing C_j relatively to the principal components of ϕ_{j-1}, or equivalently, the kernel principal components of x with respect to k_{j-1}. This defines unsupervised approximations of the weight covariance C_j by considering its projection on these first principal components. We now evaluate the quality of this approximation.

Here, we consider a seven-hidden-layer scattering network trained on CIFAR-10, and weight covariances estimated from $N = 50$ same-width networks. The upper panels of Figure 7.6 shows the amount of variance in C_j captured by the first m basis directions as a function of m, for three different orthogonal bases. The speed of growth of this variance as a function of m defines

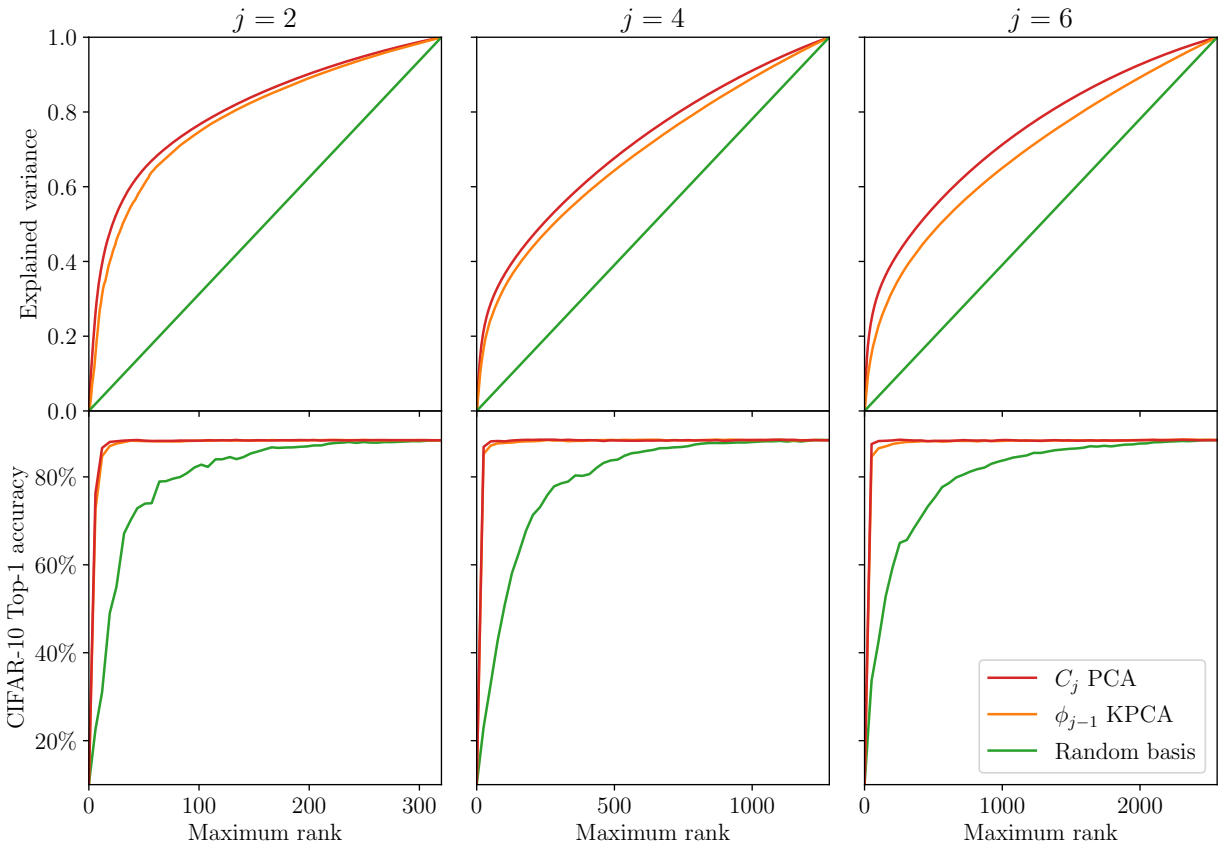

FIGURE 7.6: Unsupervised information defines low-dimensional approximations of the learned weight covariances. Each column shows a different layer $j = 2, 4, 6$ of a seven-hidden-layer scattering network trained on CIFAR-10. For each r, we consider projections of the network weights on the first r principal components of the weight covariances (red), the kernel principal components of the input activations (orange), or random orthogonal vectors (green). Top*: weight variance explained by the first r basis vectors as a function of r.* Bottom*: classification accuracy after projection of the j-th layer weights on the first r basis vectors, as function of r.*

the quality of the approximation: a faster growth indicates that the basis provides an efficient low-dimensional approximation of the covariance. The PCA basis of C_j provides optimal such approximations, but it is not known before supervised training. In contrast, the KPCA basis is computed from the previous layer activations ϕ_{j-1} without the supervision of class label information. Figure 7.6 demonstrates that the ϕ_{j-1} KPCA basis provides close to optimal approximations of C_j. This approximation is more effective for earlier layers, indicating that the supervised information becomes more important for the deeper layers. The lower panels of Figure 7.6 show a similar phenomenon when measuring classification accuracy instead of weight variance.

In summary, the learned weight matrices are low-rank, and a low-dimensional bottleneck can be introduced without harming performance. Further, unsupervised information (in the form of a KPCA) gives substantial prior information on this bottleneck: high-variance components of the weights are correlated with high-variance components of the activations. This observation was indirectly made by Raghu et al. (2017), who showed that network activations can be projected on stable subspaces, which are in fact aligned with the high-variance kernel principal components. It demonstrates the importance of self-supervised learning within supervised learning tasks (Bengio, 2012), and corroborates the empirical success of self-supervised pre-training for many supervised tasks. The effective number of parameters that need to be learned in a supervised manner is thus much smaller than the total number of trainable parameters.

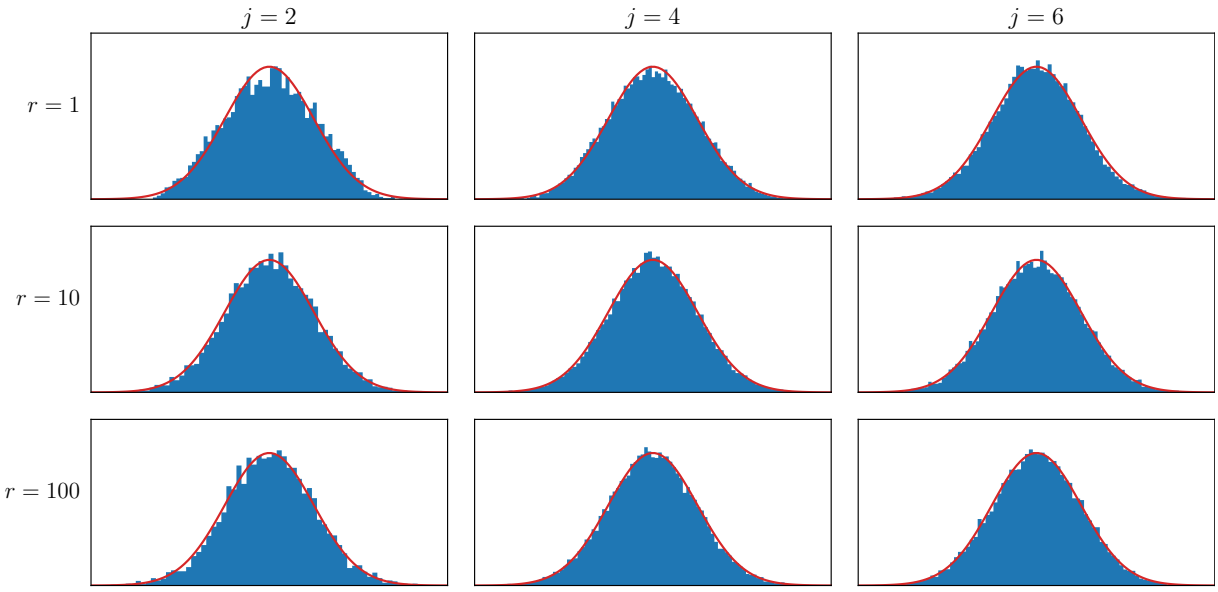

FIGURE 7.7: Marginal distributions of the weights of $N = 50$ seven-hidden-layer scattering networks trained on CIFAR-10. The weights at the j-th layer $(w_{ji})_{i \leq d_j}$ of the N networks are projected along the r-th eigenvector of C_j and normalized by the square root of the corresponding eigenvalue. The distribution of the Nd_j projections (blue histograms) is approximately normal (red curves). Each column shows a different layer j, and each row shows a different rank r.

7.3.3 Gaussian rainbow approximations

We now show that the Gaussian rainbow model applies to scattering networks trained on the CIFAR-10 dataset, by exploiting the fixed wavelet spatial filters incorporated in the architecture. The Gaussian assumption thus only applies to weights along channels. We make use of the factorization $W_j = G_j \hat{C}_j^{1/2}$ (7.13) of trained weights, where $\hat{C}_j$ results from an estimation of C_j from several trained networks. We first show that the distribution of G_j can be approximated with random matrices of i.i.d. normal coefficients. We then show that Gaussian rainbow networks, which replace G_j with such a white Gaussian matrix, achieve similar classification accuracy as trained networks when the width is large. Finally, we show that in the same context, the SGD training dynamics of the weight matrices W_j are characterized by the evolution of the weight covariances $\hat{C}_j$ only, while G_j remains close to its initial value. The Gaussian approximation deteriorates at small widths or on more complex datasets, suggesting that its validity regime is when the network width is large compared to the task complexity.

Comparison between trained weights and Gaussian matrices. We show that statistics of trained weights are reasonably well approximated by the Gaussian rainbow model. To do so, we train $N = 50$ seven-hidden-layer scattering networks and estimate weight covariances $(C_j)_{j \leq J}$ by averaging eq. (7.15) over the trained networks as explained in Section 7.3.2. We then retrieve $G_j = W_j \hat{C}_j^{-1/2}$ with $\hat{C}_j = \hat{A}_j^{\mathrm{T}} C_j \hat{A}_j$ as in eq. (7.12). Note that we use a single covariance C_j to whiten the weights of all N networks: this will confirm that the covariances of weights of different networks are indeed related through rotations, as was shown in Section 7.3.2 through the convergence of weight covariance estimates. The rainbow feature vectors $(\phi_j)_{j \leq J}$ at each layer are approximated with the activations of one of the N networks.

As a first (partial) Gaussianity test, we compare marginal distributions of whitened weights G_j with the expected normal distribution in Figure 7.7. We present results for a series of layers $(j = 2, 4, 6)$ across the network. Other layers present similar results, except for $j = 1$ which has more significant deviations from Gaussianity (not shown), as its input dimension is constrained

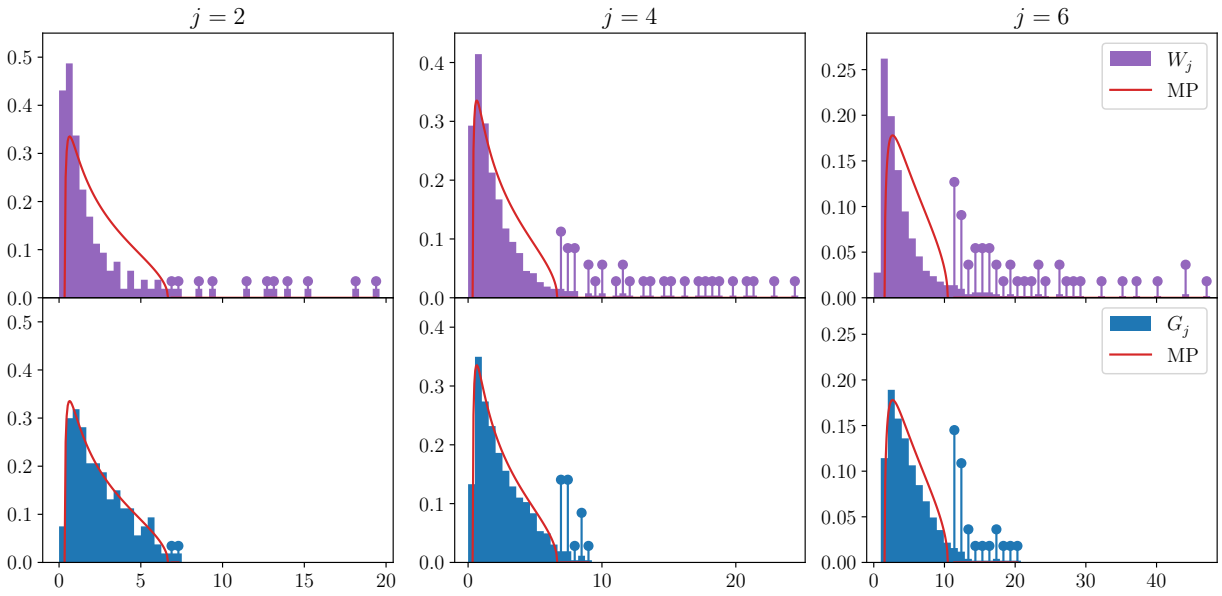

FIGURE 7.8: Spectral density of empirical covariances of trained (top) *and whitened weights* (bottom)*. Eigenvalue outside the support of the Marchenko-Pastur distribution (shown in red) are indicated with spikes of amplitude proportional to their bin count. After whitening, the number of outliers are respectively* 2%*,* 4%*, and* 8% *for the layers* $j = 2$*,* 4*, and* 6*.*

by the data dimension. We shall however not focus on this first layer as we will see that it can still be replaced by Gaussian realizations when generating new weights. The weights at the j-th layer $(w_{ji})_{i \leq d_j}$ of the N networks are projected along the r-th eigenvector of C_j and normalized by the square root of the corresponding eigenvalue. This global view shows that specific one-dimensional marginals are reasonably well approximated by a normal distribution. We purposefully remain not quantitative, as the goal is not to demonstrate that trained weights are statistically indistinguishable from Gaussian realizations (which is false), but to argue that the latter is an acceptable model for the former.

To go beyond one-dimensional marginals, we now compare in the bottom panels of Figure 7.8 the spectral density of the whitened weights G_j to the theoretical Marchenko-Pastur distribution (Marčenko and Pastur, 1967), which describes the limiting spectral density of matrices with i.i.d. normal entries. We note a good agreement for the earlier layers, which deteriorates for deeper layers (as well as the first layer, not shown, which again has a different behavior). Importantly, the proportion of eigenvalues outside the Marchenko-Pastur support is arguably negligible ($<$ 10% at all layers), which is not the case for the non-whitened weights W_j (upper panels) where it can be $> 25\%$ for $j = 6$. As observed by Martin and Mahoney (2021) and Thamm et al. (2022), trained weights have non-Marchenko-Pastur spectral statistics. Our results show that these deviations are primarily attributable to correlations introduced by the non-identity covariance matrices C_j, as opposed to power-law distributions as hypothesized by Martin and Mahoney (2021). We however note that due to the universality of the Marchenko-Pastur distribution, even a perfect agreement is not sufficient to claim that trained networks have conditionally Gaussian weights. It merely implies that the Gaussian rainbow model provides a satisfactory description of a number of weight statistical properties. Despite the observed deviations from Gaussianity at later layers, we now show that generating new Gaussian weights at all layers simultaneously preserves most of the classification accuracy of the network.

Performance of Gaussian rainbow networks. While the above tests indicate some level of validation that the whitened weights G_j are matrices with approximately i.i.d. normal entries, it is not statistically feasible to demonstrate that this property is fully satisfied in high-dimensions.

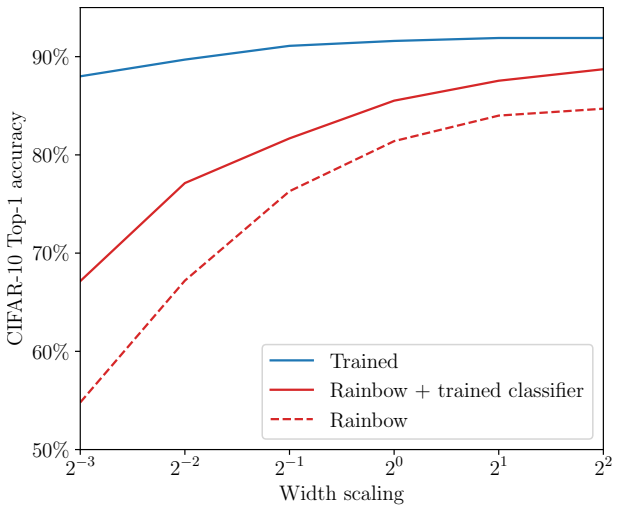

FIGURE 7.9: Performance of seven-hidden-layer scattering networks on CIFAR-10 as a function of network width for a trained network (blue), its rainbow network approximation with and without classifier retraining (red solid and dashed). The larger the width, the better the sampled rainbow model approximates the original network.

We thus sample network weights from the Gaussian rainbow model and verify that most of the performance can be recovered. This is done with the procedure described in Definition 7.2, using the covariances C_j, rainbow activations ϕ_j and final layer weights θ here estimated from a single trained network (having shown in Sections 7.3.1 and 7.3.2 that all networks define similar rainbow parameters if they are wide enough). New weights W_j are sampled iteratively starting from the first layer with a covariance $\hat{C}_j = \hat{A}_{j-1}^{\mathrm{T}} C_j \hat{A}_{j-1}$, after computing the alignment rotation $\hat{A}_{j-1}$ between the activations $\hat{\phi}_{j-1}(x)$ of the partially sampled network and the activations $\phi_{j-1}(x)$ of the trained network. The alignment rotations are computed using the CIFAR-10 train set, while network accuracy is evaluated on the test set, so that the measured performance is not a result of overfitting.

We perform this test using a series of seven-hidden-layer scattering networks trained on CIFAR-10 with various width scalings. We present results in Figure 7.9 for two sets of Gaussian rainbow networks: a first set for which both the convolutional layers and the final layer are sampled from the rainbow model (which corresponds to aligning the classifier of the trained model to the sampled activations $\hat{\phi}_J(x)$), and another set for which we retrain the classifier after sampling the convolutional layers (which preserves the Gaussian rainbow RKHS). We observe that the larger the network, the better it can be approximated by a Gaussian rainbow model. At the largest width considered here, the Gaussian rainbow network achieves 85% accuracy and 89% with a retrained classifier, and recovers most of the performance of the trained network which reaches 92% accuracy. This performance is non-trivial, as it is beyond most methods based on non-learned hierarchical convolutional kernels which obtain less than 83% accuracy (Mairal et al., 2014; Oyallon and Mallat, 2015; Li et al., 2019). This demonstrates the importance of the learned weight covariances C_j, as has been observed by Pandey et al. (2022) for modeling sensory neuron receptive fields. It also demonstrates that the covariances C_j are sufficiently well-estimated from a single network to preserve classification accuracy. We note however that Shankar et al. (2020) achieve a classification accuracy of 90% with a non-trained kernel corresponding to an infinite-width convolutional network.

A consequence of our results is that these trained scattering networks have rotation invariant non-linearities, in the sense that the non-linearity can be applied in random directions, provided that the next layer is properly aligned. This comes in contrast to the idea that neuron weights individually converge to salient features of the input data. For large enough networks, the

relevant information learned at the end of training is therefore not carried by individual neurons but encoded through the weight covariances C_j.

For smaller networks, the covariance-encoding property no longer holds, as Figure 7.9 suggests that trained weights becomes non-Gaussian. Networks trained on more complex tasks might require larger widths for the Gaussian rainbow approximation to be valid. We have repeated the analysis on scattering networks trained on the ImageNet dataset (Russakovsky et al., 2015), which reveals that the Gaussian rainbow approximation considered here is inadequate at widths used in practice. This is corroborated by many empirical observations of (occasional) semantic specialization in deep networks trained on ImageNet (Olah et al., 2017; Bau et al., 2020; Dobs et al., 2022). A promising direction is to consider Gaussian mixture rainbow models, as used by Dubreuil et al. (2022) to model the weights of linear RNNs. Finally, we note that the Gaussian approximation also critically rely on the fixed wavelet spatial filters of scattering networks. Indeed, the spatial filters learned by standard CNNs display frequency and orientation selectivity (Krizhevsky et al., 2012) which cannot be achieved with a single Gaussian distribution, and thus require adapted weight distributions π_j to be captured in a rainbow model.

Training dynamics. The rainbow model is a static model, which does not characterize the evolution of weights from their initialization during training. We now describe the SGD training dynamics of the seven-hidden-layer scattering network trained on CIFAR-10 considered above. This dynamic picture provides an empirical explanation for the validity of the Gaussian rainbow approximation.

We focus on the j-th layer weight matrix $W_j(t)$ as the training time t evolves. To measure its evolution, we consider its projection along the principal components of the final learned covariance $\hat{C}_j$. More precisely, we project the d_j neuron weights $w_{ji}(t)$, which are the rows of $W_j(t)$, in the direction of the r-th principal axis e_{jr} of $\hat{C}_j$. This gives a vector $u_r(t) \in \mathbb{R}^{d_j}$ for each PCA rank r and training time t, dropping the index j for simplicity:

$$u_r(t) = \big(\langle w_{ji}(t), e_{jr}\rangle\big)_{i \le d_j}.$$

Its squared magnitude is proportional to the variance of the neuron weights along the r-th principal direction, which should be of the order of 1 at $t = 0$ due to the white noise initialization, and evolves during training to reach the corresponding $\hat{C}_j$ eigenvalue. On the opposite, the direction of $u_r(t)$ encodes the sampling of the marginal distribution of the neurons along the r-th principal direction: a large entry $u_r(t)[i]$ indicates that neuron i is significantly correlated with the r-th principal component of $\hat{C}_j$. This view allows considering the evolution of the weights $W_j(t)$ separately for each principal component r. It offers a simpler view than focusing on each individual neuron i, because it gives an account of the population dynamics across neurons. It separates the weight matrix by columns r (in the weight PCA basis) rather than rows i. We emphasize that we consider the PCA basis of the final covariance $\hat{C}_j$, so that we analyze the training dynamics along the fixed principal axes e_{jr} which do not depend on the training time t.

We now characterize the evolution of $u_r(t)$ during training for each rank r. We separate changes in magnitude, which correspond to changes in weight variance (overall stretch), from changes in direction, which correspond to internal motions of the neurons which preserve their variance. We thus define two quantities to compare $u_r(t)$ to its initialization $u_r(0)$, namely the amplification ratio $a_r(t)$ and cosine similarity $c_r(t)$:

$$a_r(t) = \frac{\|u_r(t)\|}{\|u_r(0)\|} \quad \text{and} \quad c_r(t) = \frac{\langle u_r(t), u_r(0)\rangle}{\|u_r(t)\|\,\|u_r(0)\|}. \tag{7.16}$$

We evaluate these quantities using our seven-hidden-layer scattering network trained on CIFAR-10. In Figure 7.10, we present the results for the intermediate layer $j = 4$ (similar behavior is

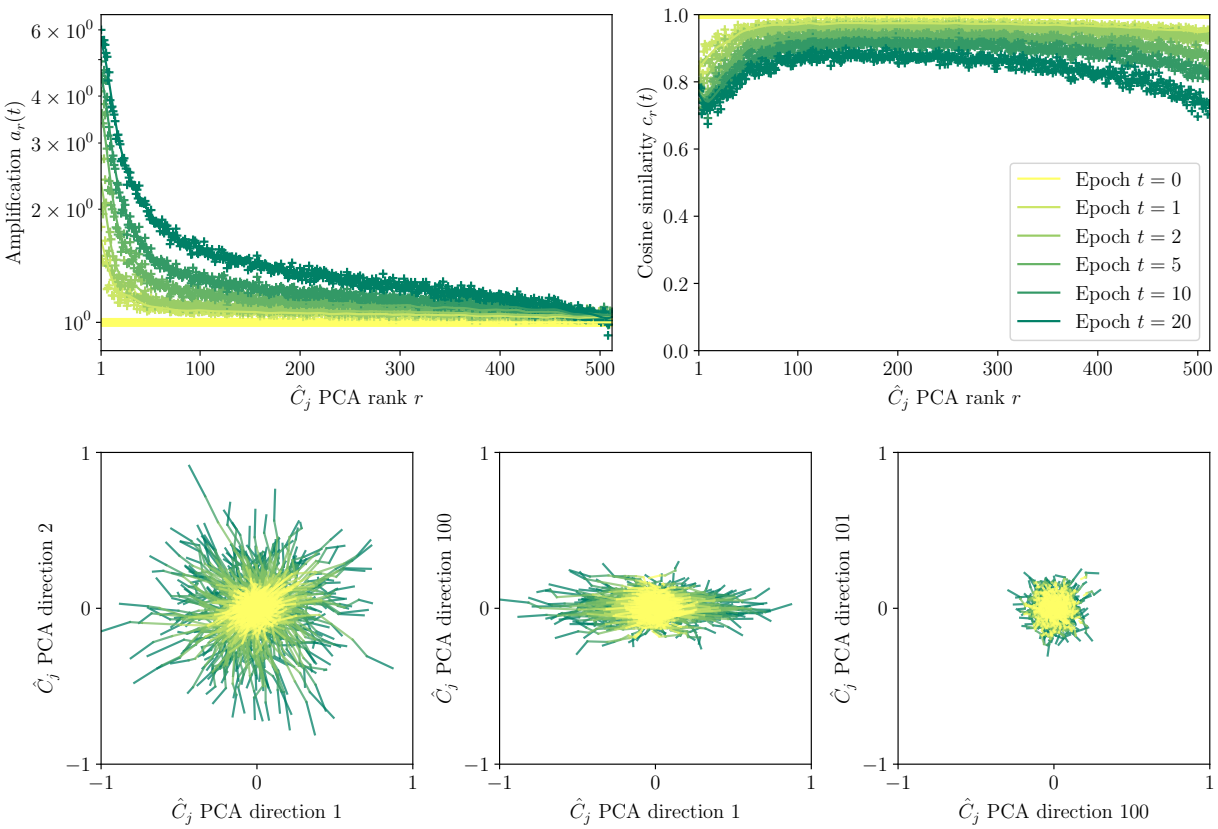

FIGURE 7.10: The learning dynamic of a seven-hidden-layer scattering network trained on CIFAR-10 is mainly a low-dimensional linear amplification effect that preserves most of the positional information of the initialization. We present results for layer $j = 4$ (similar behavior is observed for the other layers). Upper left: *amplification (overall stretch) of the weight variance as a function of rank.* Upper right: *cosine similarity (internal motion) as a function of rank.* Lower panels: *projections of individual neurons along pairs of principal components. Each neuron is represented as a point in the plane, whose trajectory during training is shown as a connected line (color indicates training time).*

observed for the other layers). We show the two quantities $a_r(t)$ and $c_r(t)$ in the top row of Figure 7.10 as a function of the training epoch t. We observe that the motion of the weight vector is mainly an amplification effect operating in a sequence starting with the first eigenvectors, as the cosine similarity remains of order unity. Given the considered dimensionality ($d_j = 512$), the observed departure from unity is rather small: the solid angle subtended by this angular change of direction covers a vanishingly small surface of the unit sphere in d_j dimensions. We thus have $u_r(t) \approx a_r(t)\, u_r(0)$.

These results show that the weight evolution can be written

$$W_j(t) \approx G_j\, \hat{C}_j^{1/2}(t),$$

where $G_j = W_j(0)$ is the initialization and the weight covariance $\hat{C}_j(t)$ evolves by amplification in its fixed PCA basis:

$$\hat{C}_j(t) = \sum_r a_r(t)^2\, e_{jr} e_{jr}^{\mathrm{T}}.$$

In other words, the weight evolution during training is an ensemble motion of the neuron population, with negligible internal motion of individual neurons relative to the population: training amounts to learning the weight covariance. Surprisingly, the weight configuration at the end of training thus retains most of the information of its random initialization: the initial configuration can be practically recovered by whitening the trained weights. In addition, the stochasticity introduced by SGD and data augmentation appears to be negligible, as it does not affect the

relative positions of individual neurons during training. This observation has two implications. First, the alignment rotations $\hat{A}_j$ which describe the trained network relative to its infinite-width rainbow counterpart (as $\hat{\phi}_j \approx \hat{A}_j^{\mathrm{T}} \phi_j$) are entirely determined by the initialization. Second, it provides an empirical explanation for the validity of the Gaussian rainbow approximation. While this argument seems to imply that the learned weight distributions π_j depend significantly on the initialization scheme, note that significantly non-Gaussian initializations might not be preserved by SGD or could lead to poor performance.

The bottom row of Figure 7.10 illustrates more directly the evolution of individual neurons during training. Although each neuron of $W_j(t)$ is described by a d_{j-1}-dimensional weight vector, it can be projected along two principal directions to obtain a two-dimensional picture. We then visualize the trajectories of each neuron projected in this plane. The trajectories are almost straight lines, as the learning dynamics only amplify variance along the principal directions while preserving the relative positions of the neurons. Projections on principal components of higher ranks give a more static picture as the amplification along these directions is smaller.

A large literature has characterized properties of SGD training dynamics. Several works have observed that dynamics are linearized after a few epochs (Jastrzebski et al., 2020; Leclerc and Madry, 2020), so that the weights remain in the same linearly connected basin thereafter (Frankle et al., 2020). It has also been shown that the empirical neural tangent kernel evolves mostly during this short initial phase (Fort et al., 2020) and aligns itself with discriminative directions (Baratin et al., 2021; Atanasov et al., 2022). Our results indicate that this change in the neural tangent kernel is due to the large amplification of the neuron weights along the principal axes of $\hat{C}_j$, which happen early during training. The observation that neural network weights have a low-rank departure from initialization has been made in the lazy regime by Thamm et al. (2022), for linear RNNs by Schuessler et al. (2020), and for large language-model adaptation by Hu et al. (2022). The sequential emergence of the weight principal components has been derived theoretically in linear networks by Saxe et al. (2014, 2019).

7.4 Discussion

We have introduced rainbow networks as a model of the probability distribution of weights of trained deep networks. The rainbow model relies on two assumptions. First, layer dependencies are reduced to alignment rotations. Second, neurons are independent when conditioned on the previous layer weights. Under these assumptions, trained networks converge to a deterministic function in the corresponding rainbow RKHS when the layer widths increase. We have verified numerically the convergence of activations after alignment for scattering networks and ResNets trained on CIFAR-10 and ImageNet. We conjecture that this convergence conversely implies the rotation dependency assumption of the rainbow model. We have verified this rotation on the second-order moments of the weights through the convergence of their covariance after alignment (for scattering networks trained on CIFAR-10 due to computational limitations).

The data-dependent kernels which describe the infinite-width rainbow networks, and thus their functional properties, are determined by the learned distributions π_j. Mathematically, we have shown how the symmetry properties of these distributions are transferred on the network. Numerically, we have shown that their covariances C_j compute projections in a low-dimensional "informative" subspace that is shared among networks, is low-dimensional, and can be approximated efficiently with an unsupervised KPCA. It reveals that networks balance low learning complexity with high expressivity by computing a sequence of reductions and increases in dimensionality.

In the Gaussian case, the distributions π_j are determined by their covariances C_j. We have validated that factorizing the learned weights with fixed wavelet filters is sufficient to obtain Gaussian rainbow networks on CIFAR-10, using scattering networks. In this setting, we can generate new weights and have shown that the weight covariances C_j are sufficient to capture

most of the performance of the trained networks. Further, the training dynamics are reduced to learning these covariances while preserving memory of the initialization in the individual neuron weights.

Our work has several limitations. First, we have not verified the rainbow assumptions of rotation dependence between layers beyond second-order moments, and conditional independence between neurons beyond the Gaussian case. A complete model would incorporate the training dynamics and show that such statistical properties are satisfied at all times. Second, our numerical experiments have shown that the Gaussian rainbow approximation of scattering networks gradually degrades when the network width is reduced. When this approximation becomes less accurate, it raises the question whether incorporating more prior information in the architecture could lead to Gaussian rainbow networks. Finally, even in the Gaussian case, the rainbow model is not completely specified as it requires to estimate the weight covariances C_j from trained weights. A major mathematical issue is to understand the properties of the resulting rainbow RKHS which result from properties of these weight covariances.

By introducing the rainbow model, this chapter provides new insights towards understanding the inner workings of deep networks. It leads to a conjecture for a static mean-field limit of deep networks presented in Section 1.4.3, which may be used to study their training dynamics.

Conclusion

Chapter

8 Conclusion

Les ogres sont parfois poètes.

La magicienne

Chapter content

8.1 Summary of findings **129**

8.2 Perspectives **130**

In this dissertation, I have tried to uncover the hidden mathematical structure in image distributions, deep convolutional architectures, and deep network weights. A guiding principle has been to separate the problem across scales and exploit the spatial structure of images, which performs a reduction to problems along channels only. However, much more remains to be done, and in particular integrating the results presented in this dissertation in a unified picture.

What does it even mean to understand what the network has learned and what it is computing? It calls for a better understanding of the relationships between properties of the training data, network weights, and hidden activations. Studying unsupervised and supervised learning together might be a step towards this goal and may lead to fruitful interactions in the future.

We summarize our findings in Section 8.1, and end the dissertation with a few perspectives for future research outlined in Section 8.2.

8.1 Summary of findings

Multiscale conditional probabilities. In Chapters 2 to 4, we have established several properties of the wavelet (packet) conditional distributions $p(\bar{x}_j|x_j)$ in different contexts. Namely, these conditional distributions are log-concave for multiscale physical fields whose interactions are dominated by a quadratic kinetic energy at high-frequencies. It provides a wider class of probability distributions than globally log-concave distributions that still allows breaking the curse of dimensionality for both learning and sampling. We have also initiated a study of the properties of the scores of natural image distributions. We have shown that a multiscale factorization leads to approximate stationarity and locality, but one can expect that there is much more to uncover.

Role of non-linearity in image classification. In Chapters 5 and 6, we have investigated the role of the non-linearity in image classification architectures. We have shown that its main function is to collapse the phase of hidden network activations, as opposed to computing sparse representations with a thresholding. This role can be further constrained to collapsing the phase of wavelet coefficients, leading to a structured architecture which relies solely on this mechanism. It allows eliminating biases and pre-defining all spatial filters, and thus learning only weights along channels. This constrained architecture strikes a good balance between performance and amenability to analysis, and thus might be useful for future research.

Structure of weights in deep networks. In Chapter 7, we have identified that network activations define an alignment that can be used to register the next layer weights. It has led us to a conjecture for a multi-layer mean-field limit of deep neural networks. This view also gives insights into the behavior of deep networks, revealing that they compute an alternating sequence of operators that increase or decrease the dimensionality of intermediate representations. Intriguingly, the training dynamics appear straightforward in relatively simple cases, which calls for a deeper explanation. Combined with our learned scattering architecture, we obtained a probabilistic model of trained weights with conditionally Gaussian distributions.

The rainbow model demonstrates the efficiency of random projections as a means to compute an approximate dot-product kernel embedding. Importantly, it highlights the importance of the covariance of the random features, and explains that this covariance needs to be aligned to the previous layer activations if such operators are cascaded. This non-linear operator is of a different nature than the phase collapses. Colored random projections, which are reminiscent of compressed sensing, are thus a new tool which may also prove useful in image generative modeling.

8.2 Perspectives

Spatial and probability scales. It is interesting to draw parallels between the wavelet conditional factorization, which iteratively generates images at different scales $(x_J, \dots, x_0)$, and a diffusion model, which iteratively generates images at different noise levels $(x_T, \dots, x_0)$.

The wavelet conditional factorization represents a distribution by conditional energies at each scale, or conditional scores when there is log-concavity, which may not always be the case. This representation avoids the need to deal with unstable free energies, and leverages a self-similarity over scales to obtain low-dimensional parameterized models.

In diffusions models, we get log-concavity "for free", and we represent a distribution by its scores at all noise levels. It provides a new way to think about and parameterize probability distributions. The noise level axis can be thought of as a probability scale axis, thus painting a geometric picture: the scores $(\nabla E_t(x))_{t,x}$ are a scale-space representation of the original probability distribution through (soft) "projection" operators on its support. It is a central issue to understand the properties of this probability scale space and its relationship with the more classical image scale space. In particular, comparing the two raises the question whether there is a form of self-similarity properties of scores across noise levels that can be exploited to reduce the dimensionality of parametric models of such scores.

Probability decompositions. More generally, one could consider general latent variables $(x = z_0, z_1, \dots, z_n)$ such that the forward conditionals $p(z_i|z_0, \dots, z_{i-1})$ are easy to sample from, and learn models of the backward conditional distributions $p(z_i|z_{i+1}, \dots, z_n)$. The wavelet conditional factorization and diffusion models are both special cases of this more general framework, with explicit forward probability distributions that are Markov and either a delta function or a Gaussian distribution. One can wonder whether there are other useful such decompositions with other latent variables, what would be their mathematical structure, or whether one could learn them from data. One example that would be worth exploring is to consider the activations $(\phi_j(x))_j$ of a pre-trained deep network, such as an image classifier. Desirable properties of these decompositions include having log-concave (or even Gaussian) backward conditional distributions that admit low-dimensional parametric models, while minimizing the number and dimensionality of the latent variables for computational efficiency. Such decompositions seem to be a powerful tool to factorize probability distributions into tractable factors while avoiding the instability issues that can arise from directly learning energy functions of distributions near a "critical point".

Data, activations, and weights. Just as the activations of a deep network can be thought of as latent variables, the weights of this network are a parameterization of the data probability distribution $E(x)$ or $E(y|x)$. Understanding the mathematical relationships between these three objects (activations, weights, and data) is a central issue to resolve the deep learning mystery, and it is fruitful to consider them together and study their relationships. For instance, what are the weight distributions in score networks, and can we relate them to properties of the data distribution? How do the weight distributions at a given layer depend on the activations at the previous layer, and how do these weight distributions in turn give rise to the activations at the next layer? Answering these questions would lead to a much more complete picture of the behavior of deep networks, their computations, and what they have learned.

Dimensionality and function classes. We observed that activations are high-dimensional (their spectrum follows a power-law of index close to negative unity) while weights are low-dimensional (their spectrum is approximately exponential with a relatively small characteristic scale), but we have not addressed how and why this is the case. These spectra are probably related to a measure of the "size" or "complexity" of the associated reproducing kernel Hilbert spaces, and therefore properties of the approximation class including generalization performance.

Training dynamics and algorithms. Another question we left open is to model the training dynamics of deep networks: can we prove a multi-layer mean-field limit using alignment to feature vectors associated to time-dependent kernels? We made the surprising observation that training dynamics of learned scattering networks on the CIFAR-10 dataset are mostly "in a straight line". Mathematically, this calls for an explanation of this phenomenon, which would also explain the remarkable stability of hidden network activations across layers. Numerically, a direction for future research would be to try to leverage these observations to design architectures and training algorithms that are more time- and data-efficient, which could potentially have an important practical impact.

Understanding depth. An important limitation of the rainbow model is that it is only a layer-wise characterization of the network, and the global picture remains elusive. In particular, it is not clear how one would compare networks with different numbers of layers. A possible direction would be to derive an infinite-depth limit and define alignments of a finite-depth network to its infinite-depth counterpart. An infinite-depth network defines a continuous transport of its input to its output through a continuous sequence of intermediate representations. What are the properties of this transport? It also raises the question of the nature of the depth axis. In a CNN, depth corresponds to both spatial scale (the size of the receptive fields of a given neuron) and non-linearity order (the amount of non-linear processing done to the input). An important issue is to understand the nature and role of depth, perhaps by disentangling different notions such as scale and order.

Appendices

Appendix

A Appendix for Chapter 2

Chapter content

A.1 Definition of wavelet packet projectors . . . 135
A.1.1 Conjugate mirror filters . . . 135
A.1.2 Orthogonal frequency decomposition . . . 136
A.1.3 Wavelet packet projectors . . . 137
A.2 Score matching and MALA algorithms for CSLC exponential families 137
A.2.1 Multiscale energies . . . 137
A.2.2 Pseudocode . . . 140
A.3 Experimental details . . . 140
A.3.1 Datasets . . . 140
A.3.2 Experimental setup . . . 141
A.3.3 Mixing times in MALA . . . 141
A.4 Energy estimation with free-energy modeling . . . 142
A.4.1 Free-energy score matching . . . 142
A.4.2 Parameterized free-energy models . . . 142
A.4.3 Multiscale energy decomposition . . . 143
A.5 Proof of Proposition 2.3 . . . 144

A.1 Definition of wavelet packet projectors

The fast wavelet transform (Mallat, 1989) splits a signal in frequency into two orthogonal coarser signals, using two orthogonal conjugate mirror filters g and $\bar{g}$.

We review the construction of such filters in appendix A.1.1. A description of the fast wavelet transform is then given in appendix A.1.2. Finally, we define in appendix A.1.3 the wavelet packet (Coifman et al., 1992) projectors $(G_j, \bar{G}_j)$ used in Section 2.3.

A.1.1 Conjugate mirror filters

Conjugate mirror filters g and $\bar{g}$ satisfy the orthogonal and reconstruction conditions

$$\begin{aligned} g^{\mathrm{T}}\bar{g} = \bar{g}^{\mathrm{T}}g = 0, \\ g^{\mathrm{T}}g + \bar{g}^{\mathrm{T}}\bar{g} = \mathrm{Id}\,. \end{aligned} \tag{A.1}$$

In one dimension, the conditions (A.1) are satisfied (Mallat, 1989) by discrete filters $(g(n))_{n\in\mathbb{Z}}, (\bar{g}(n))_{n\in\mathbb{Z}}$ whose Fourier transforms $\hat{g}(\omega) = \sum_n g(n)e^{-in\omega}$ and $\hat{\bar{g}}(\omega) = \sum_n g(n)e^{-in\omega}$ satisfy

$$\begin{aligned} &|\hat{g}(\omega)|^2 + |\hat{g}(\omega+\pi)|^2 = 2, \\ &\hat{g}(0) = \sqrt{2}, \\ &\hat{\bar{g}}(\omega) = e^{-i\omega}\hat{g}(\omega+\pi). \end{aligned} \tag{A.2}$$

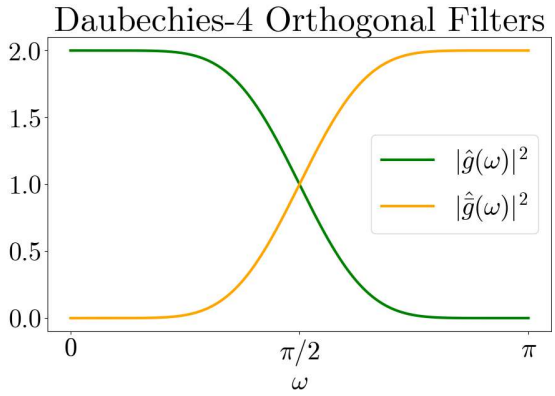

FIGURE A.1: Fourier transform of Daubechies-4 orthogonal filters $\hat{g}(\omega)$ (in green) and $\hat{\bar{g}}(\omega)$ (in orange).

We first design a low-frequency filter g such that $\hat{g}(\omega)$ satisfies (A.2), and then compute $\bar{g}$ with

$$\bar{g}(n) = (-1)^{1-n} g(1-n). \tag{A.3}$$

The choice of a particular low pass filter g is a trade-off between a good localization in space and a good localization in the Fourier frequency domain. Choosing a perfect low-pass filter $g(\omega) = \mathbb{1}_{\omega \in [-\pi/2, \pi/2]}$ leads to Shannon wavelets, which are well localized in the frequency domain but have a slow decay in space. On the opposite, a Haar wavelet filter $g(n) = \sqrt{2}\,\mathbb{1}_{n \in \{0,1\}}$ has a small support in space but is poorly localized in frequency. Daubechies filters (Daubechies, 1992) provide a good joint localization both in the spatial and Fourier domains. The Daubechies-4 wavelet is shown in Figure A.1.

In two dimensions (for images), wavelet filters which satisfy the orthogonality conditions in (A.1) can be defined as separable products of the one-dimensional filters g and $\bar{g}$ (Mallat, 2008), applied on each coordinate. It defines one low-pass filter g_2 and 3 high-pass filters $\bar{g}_2 = (\bar{g}_2^k)_{1 \leq k \leq 3}$:

$$\begin{aligned} g_2(n_1, n_2) &= g(n_1) g(n_2), \\ \bar{g}_2^1(n_1, n_2) &= g(n_1) \bar{g}(n_2), \\ \bar{g}_2^2(n_1, n_2) &= \bar{g}(n_1) g(n_2), \\ \bar{g}_2^3(n_1, n_2) &= \bar{g}(n_1) \bar{g}(n_2). \end{aligned} \tag{A.4}$$

For simplicity we shall write g and $\bar{g}$ the filters g_2 and $\bar{g}_2$. $\bar{g}$ outputs the concatenation of the 3 filters $\bar{g}_2^k$.

A.1.2 Orthogonal frequency decomposition

We introduce the orthogonal decomposition of a signal x_{j-1} with the low pass filter g and the high pass filter $\bar{g}$, followed by a sub-sampling. It outputs $(x_j, \bar{x}_j)$, which has the same dimension as x_{j-1}, defined in one dimension by

$$\begin{aligned} x_j[p] &= \sum_{n \in \mathbb{R}^2} g[n - 2p] x_{j-1}[n], \\ \bar{x}_j[p] &= \sum_{n \in \mathbb{R}^2} \bar{g}[n - 2p] x_{j-1}[n]. \end{aligned} \tag{A.5}$$

The inverse transformation is

$$x_j[p] = \sum_{n \in \mathbb{R}^2} g[p - 2n] x_{j+1}[n] + \sum_{n \in \mathbb{R}^2} \bar{g}[p - 2n] \bar{x}_{j+1}[n]. \tag{A.6}$$

The orthogonal frequency decomposition in two dimensions is defined similarly. It decomposes a signal x of size $\sqrt{d} \times \sqrt{d}$ into a low frequency signal and 3 high frequency signals, each of size $\frac{\sqrt{d}}{2} \times \frac{\sqrt{d}}{2}$.

A.1.3 Wavelet packet projectors

An orthogonal frequency decomposition projects a signal into high and low frequency domains. In order to refine the decomposition (by separating different frequency bands), wavelet packets projectors are obtained by cascading this orthogonal frequency decomposition.

The usual fast wavelet transform starts from a signal $\bar{x}_0$ of dimension d, decomposes it into a low-frequency x_1 and a high frequency $\bar{x}_1$, and then iterates this decomposition on the low-frequency x_1 only. It iteratively decomposes x_{j-1} into the lower frequencies x_j and the high-frequencies $\bar{x}_j$. The resulting orthogonal wavelet coefficients are $(\bar{x}_j, x_J)_{1\leq j\leq J}$. The resulting decomposition remains of dimension d.

To obtain a finer frequency decomposition, we use the M-band wavelet transform (Mallat, 2008), a particular case of wavelet packets (Coifman et al., 1992). It first applies the fast wavelet transform to the signal, and obtains $(\bar{x}_j, x_J)_{1\leq j\leq J}$. Each high-frequency output $\bar{x}_j$ undergoes an orthogonal decomposition using g and $\bar{g}$. Then both outputs of the decomposition are again decomposed, and so on, $(M-1)$-times. The coefficients are then sorted according to their frequency support, and also labeled as $(\bar{x}_j, x_J)_{1\leq j\leq J'}$, with $J' = J2^{M-1}$, also referred to as J in the main text.

The wavelet packet decomposition corresponds to first decomposing the frequency domain dyadically into octaves, and then each dyadic frequency band is further decomposed into 2^{M-1} frequency annuli. We say this decomposition corresponds to a $1/2^{M-1}$ octave bandwidth. Precisely, if $j = j'2^{M-1} + r$, then $\bar{x}_j$ has a frequency support over an annulus in the frequency domain, with frequencies with modulus of order $2^{-j'}\pi(1 - 2^{-M+1}(r - 1/2))$. A two-dimensional visualization of the frequency domain can be found in Figure 2.2, for $M = 1$ and $M = 2$, corresponding to 1 and 1/2 octave bandwidths.

Figure A.2 shows the iterative use of g and $\bar{g}$ used to obtain the decomposition, in one dimension, for $M = 2$. Note that the filters $\bar{g}$ and g successively play the role of low- and high-pass filters because of the subsampling (Mallat, 2008).

We now introduce the corresponding orthogonal projectors G_j and $\bar{G}_j$, defined such that

$$\begin{aligned} \bar{x}_j &= \bar{G}_j x_{j-1}, \\ x_j &= G_j x_{j-1}, \end{aligned} \tag{A.7}$$

where the $(\bar{x}_j)_j$, sorted in frequency, have been obtained trough the M-band wavelet transform, as described above, and x_j refers to the signal reconstructed using $(x_J, \bar{x}_{j'})_{j'\geq j+1}$. Let us emphasize that the image x_{j-1} is reconstructed from x_j and the higher frequencies $\bar{x}_j$, and defined on a spatial grid which is either the same as x_j or twice larger. For $M = 2$, Figure A.3 shows that x_0 and x_1 are defined on the same grid, although x_1 has a lower-frequency support. Similarly x_2 and x_3 are both represented on the same grid, which is twice smaller, and so on.

The orthogonal projectors satisfy $G_j^{\mathrm{T}} G_j + \bar{G}_j^{\mathrm{T}} \bar{G}_j = \mathrm{Id}$. We then have the inverse formula

$$x_{j-1} = G_j^{\mathrm{T}} x_j + \bar{G}_j^{\mathrm{T}} \bar{x}_j. \tag{A.8}$$

This decomposition using G_j and $\bar{G}_j$ recursively splits the signal in frequencies, from high to low frequencies.

A.2 Score matching and MALA algorithms for CSLC exponential families

A.2.1 Multiscale energies

This section introduces the explicit parametrization of the energies $\bar{E}_{\bar{\theta}_j}$ and E_{θ_J}.

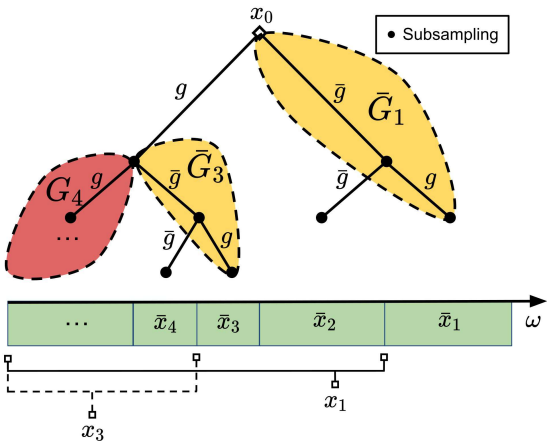

FIGURE A.2: In one dimension, a wavelet packet transform is obtain by cascading filterings and subsamplings with the filters g and $\bar{g}$ along a binary splitting tree which outputs x_J and $\bar{x}_j$ for $j \geq J$.

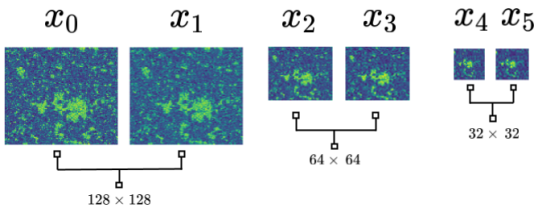

FIGURE A.3: Low-frequency maps x_j for $M = 2$ for a φ^4 realization.

The conditional energies $\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j)$ are defined with a bilinear term which represents the interaction between x_j and $\bar{x}_j$ and a scalar potential:

$$\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j) = \frac{1}{2}\bar{x}_j^{\mathrm{T}} \bar{K}_j \bar{x}_j + \sum_{l>j} \bar{x}_j^{\mathrm{T}} \bar{K}'_{l,j} \bar{x}_{j+l} + \sum_i \bar{v}_j(x_{j-1}[i]), \tag{A.9}$$

with $x_{j-1} = \bar{G}_j^{\mathrm{T}} \bar{x}_j + G_j^{\mathrm{T}} x_j$. Equation (A.9) is an equivalent reparametrization of eq. (2.13). Considering $(\bar{x}_l)_{l>j}$ instead of x_j allows fixing some coefficients of the $\bar{K}'_{l,j}$ to zero instead of learning them. First, we set $\bar{K}'_{l,j} = 0$ if $\bar{x}_j$ and $\bar{x}_{j+l}$ are not defined on the same spatial grid. In the sequel, sums over l only refer to theses terms, which differ depending on the wavelet decomposition. We enforce spatial stationarity by averaging the bilinear interaction terms across space. We further kept only the non-negligible terms which correspond to neighboring frequencies and neighboring spatial locations. As displayed in Figure A.4, $\bar{x}_j$ is composed of sub-bands $\bar{x}_j^k$. We kept the interaction terms $\bar{x}_j^k[i]\bar{x}_{j+l}^{k+\delta k}[i + \delta i]$ for $l \in \{0, 1\}$, $\delta k \in \{0, 1\}$, and $\delta i \in \{0, 1, 2, 3, 4\}^2$, which correspond to local interactions in both space and frequency.

The scalar potential $\bar{v}_j(t)$ is decomposed on a family of predefined functions $\rho_{k,j}(t)$:

$$\bar{v}_j(t) = \sum_k \bar{\alpha}_{k,j}\, \rho_{k,j}(t). \tag{A.10}$$

$\rho_{j,k}$ is defined in order to expand the scalar potential $\bar{v}_j$ which captures the marginal distributions of the $x_{j-1}[i]$, which do not depend on i due to stationarity. We divide this marginal into N quantiles. Each $\rho_{k,j}$ is chosen to be a regular bump function having a finite support on the k-th quantile. This parametrization performs a pre-conditioning of the score matching Hessian.

Let ρ be a bump function with a support in $[-1/2, 1/2]$. For each j, let $a_{j,k}$ and $l_{j,k}$ be respectively the center and width of the k-th quantile of the marginal distribution of $\bar{x}_j$, we

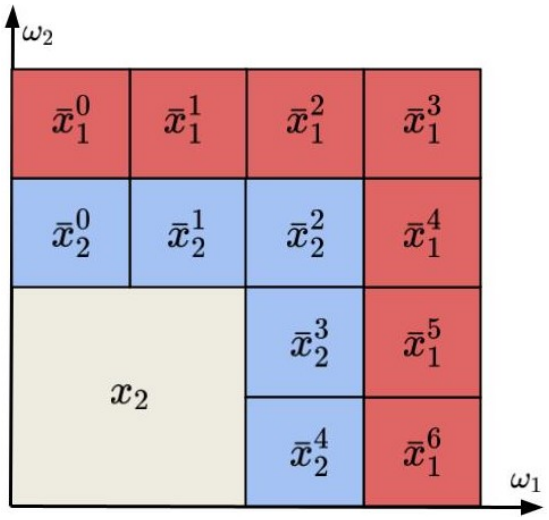

Figure A.4: Sub-bands of $\bar{x}_j$ for a wavelet packet decomposition with a half-octave bandwidth.

define

$$\rho_{k,j}(t) = l_k \sqrt{N}\, \rho\left(\frac{t - a_{j,k}}{l_{j,k}}\right), \tag{A.11}$$

with the condition

$$\|\rho'\|_2^2 = \frac{1}{\|\bar{G}_j\|_2^2}, \tag{A.12}$$

in order to balance the magnitude of the scalar potentials with the quadratic potentials.

The potential vector is thus

$$\bar{\Phi}_j(x_j, \bar{x}_j) = \left(\sum_i \bar{x}_j^k[i]\bar{x}_{j+l}^{k+\delta k}[i+\delta_i], \sum_i \rho_{k',j}(x_{j-1}[i])\right)_{0\le l\le 1, 0\le \delta k\le 1, 0\le \delta_i\le 4, 1\le k'\le N}. \tag{A.13}$$

Similarly, we define E_{θ_J} as the sum of a quadratic energy and a scalar potential:

$$E_{\theta_J}(x_J) = \frac{1}{2} x_J^{\mathrm{T}} K_J x_J + \sum_i v_J(x_J[i]). \tag{A.14}$$

The bilinear interaction terms are averaged across space to enforce stationarity. The scalar potential $v_J(t)$ is also decomposed over a family of predefined functions $\rho_{k,J}(t)$:

$$v_J(t) = \sum_k \alpha_{k,J}\, \rho_{k,J}(t), \tag{A.15}$$

defined similarly as above. This yields a potential vector

$$\Phi_J(x_J) = \left(\sum_i x_J[i] x_J[i+\delta_i], \rho_{k,J}(x_J)\right)_{0\le \delta i\le 4, 1\le k\le N}, \tag{A.16}$$

leading to

$$E_{\theta_J}(x_J) = \theta_J^{\mathrm{T}} \Phi_J(x_J), \tag{A.17}$$

with $\theta_J = (K_J, \alpha_{k,J})_k$.

A.2.2 Pseudocode

The procedure to learn the parameters $(\bar{\theta}_j)_j$ of the conditional energies $\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j)$ by score matching is detailed in Algorithm A.1. The procedure to generate samples from the distribution $p_\theta(x)$ with MALA is detailed in Algorithm A.2.

Algorithm A.1 Score matching for exponential families with CSLC distributions

Require: Training samples $(x^i)_{1 \leq i \leq n}$.
Initialize $x_0^i = x^i$ for $1 \leq i \leq n$.
for $j = 1$ **to** J **do**
Decompose $x_j^i \leftarrow G_j x_{j-1}^i$ and $\bar{x}_j^i \leftarrow \bar{G}_j x_{j-1}^i$ for $1 \leq i \leq n$.
Compute the score matching quadratic term $H_j \leftarrow \frac{1}{n} \sum_{i=1}^n \nabla_{\bar{x}_j} \bar{\Phi}_j(x_j^i, \bar{x}_j^i) \nabla_{\bar{x}_j} \bar{\Phi}_j(x_j^i, \bar{x}_j^i)^{\mathrm{T}} \in \mathbb{R}^{m \times m}$.
Compute the score matching linear term $g_j \leftarrow \frac{1}{n} \sum_{i=1}^n \Delta_{\bar{x}_j} \bar{\Phi}_j(x_j^i, \bar{x}_j^i) \in \mathbb{R}^m$.
Set $\bar{\theta}_j \leftarrow H_j^{-1} g_j$.
end for
return Model parameters $(\bar{\theta}_j)_j$.

Algorithm A.2 MALA sampling from CSLC distributions

Require: Model parameters $(\bar{\theta}_j)_j$, an initial sample x_J from $p(x_J)$, step sizes $(\delta_j)_j$, number of steps $(T_j)_j$.
for $j = J$ **to** 1 **do**
Initialize $\bar{x}_{j,0} = 0$.
for $t = 1$ **to** T_j **do**
Sample $\bar{y}_{j,t} \sim \mathcal{N}\Big(\bar{x}_{j,t-1} - \delta_j \nabla_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_{j,t-1}), 2\delta_j \,\mathrm{Id}\Big)$.
Set $a = \Big\|\nabla_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{y}_{j,t}))\Big\|^2 + \Big\|\nabla_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_{j,t-1}))\Big\|^2$.
Set $b = \Big\langle \bar{y}_{j,t} - \bar{x}_{j,t-1}, \nabla_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{y}_{j,t}) - \nabla_{\bar{x}_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_{j,t-1}) \Big\rangle$.
Set $c = \bar{E}_{\bar{\theta}_j}(x_j, \bar{y}_{j,t}) - \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_{j,t-1})$.
Compute acceptance probability $p = \exp\Big(-\frac{\delta_j}{4} a + \frac{1}{2} b - c\Big)$.
Set $\bar{x}_{j,t} = \bar{y}_{j,t}$ with probability p and $\bar{x}_{j,t} = \bar{x}_{j,t-1}$ with probability $1 - p$.
end for
Reconstruct $x_{j-1} = G_j^{\mathrm{T}} x_j + \bar{G}_j^{\mathrm{T}} \bar{x}_{j,T_j}$.
end for
return a sample x_0 from $\hat{p}_\theta(x)$.

A.3 Experimental details

A.3.1 Datasets

Simulations of φ^4. We used samples from the φ^4 model generated using a classical MCMC algorithm, for 3 different temperatures, at the critical temperature $\beta_c \approx 0.68$, above the critical temperature at $\beta = 0.50 < \beta_c$, and below the critical temperature at $\beta = 0.76 > \beta_c$. For $\beta = 0.76$, we break the symmetry and only generate samples with positive mean. For each temperature, we generate 10^4 images of size 128×128.

Weak lensing. We used down-sampled versions of the simulated convergence maps from the Columbia Lensing Group (`http://columbialensing.org/`; Zorrilla Matilla et al., 2016; Gupta

et al., 2018). Each map, originally of size 1024×1024, is downsampled twice with local averaging. We then extract random patches of size 128×128.

To pre-process the data, we subtract the minimum of the pixel values over the entire dataset, and then take the square root. This process is reversed after generating samples. We also do not consider the outliers (less than 1% of the dataset) with pixels above a certain cutoff, in order to reduce the extent of the tail and attenuate weak lensing peaks. Our dataset is made of $\simeq 4 \times 10^3$ images.

A.3.2 Experimental setup

Wavelet filter. We used the Daubechies-4 wavelet (Daubechies, 1992), see the filter in Figure A.1.

Wavelet packets. We implemented wavelet packets in PyTorch, inspired from the PyWavelets software (Lee et al., 2019a).

Score matching. We pre-condition the score matching Hessian H_j by normalizing its diagonal before computing $H_j^{-1} g_j$ in Algorithm A.1. After this normalization, we obtain condition numbers $\kappa_{\bar{\theta}_j}$ which satisfy $\kappa_{\bar{\theta}_j} \leq 2 \times 10^3$ at all j.

Sampling. The MALA step sizes δ_j are adjusted to obtain an optimal acceptance rate of ≈ 0.57. Depending on the scale j, the stationary distribution is reached in $T_j \approx 20$–400 iterations from a white noise initialization. We used a qualitative stopping criterion according to the quality of the matching of the histograms and power spectrum.

A.3.3 Mixing times in MALA

Sampling from p_θ requires sampling from p_{θ_J}, and then conditionally sampling from $p_{\bar{\theta}_j}(\bar{x}_j | x_j)$. This last step is performed with a Markov chain whose stationary distribution is $p_{\bar{\theta}_j}(\bar{x}_j | x_j)$ for a given x_j. It generates successive samples $\bar{x}_j(t)$ where t is the the step number in the Markov chain.

We introduce the conditional auto-correlation function:

$$A_j(t) = \frac{\mathbb{E}\big[(\bar{x}_j(t) - \mathbb{E}[\bar{x}_j \,|\, x_j])(\bar{x}_j(0) - \mathbb{E}[\bar{x}_j \,|\, x_j])\big]}{\mathbb{E}[\delta \bar{x}_j^2]}.$$

The expected value $\mathbb{E}$ is taken with respect to both x_j and the sampled $\bar{x}_j$. $A_j(t)$ has an exponential decay. Let $\bar{\tau}_j$ be the mixing time defined as the time it takes for the Markov chain to generate two independent samples:

$$A_j(t) \approx A_j(0) \exp\left(-\frac{t}{\bar{\tau}_j}\right).$$

$\bar{\tau}_j$ is computed by regressing $\log(A_j(t))$ over t.

Each iteration of MALA with $p_{\bar{\theta}_j}(\bar{x}_j \mid x_j)$ computes a gradient of size $\bar{d}_j$. In order to estimate the real computational cost of the sampling of p_θ, we average $\bar{\tau}_j$ proportionally to the dimension $\bar{d}_j$:

$$\bar{\tau} = \sum_{j=1}^{J} \frac{\bar{d}_j}{d} \bar{\tau}_j + \tau_J \frac{d_J}{d},$$

where d is the dimension of x.

A.4 Energy estimation with free-energy modeling

This section explains how to recover an explicit parametrization of the negative log-likelihood $-\log p_\theta$ from the parameterized energies $\bar{E}_{\bar{\theta}_j}$. We introduce a parameterization of the normalization constant of the Gibbs energies for each j and describe an efficient score-matching algorithm to learn the parameters. This leads to a decomposition of the negative log-likelihood $-\log p_\theta$ over scales.

A.4.1 Free-energy score matching

From the decomposition

$$p_\theta(x) = p_{\theta_J}(x_J) \prod_{j=1}^{J} p_{\bar{\theta}_j}(\bar{x}_j | x_j),$$

we obtain

$$-\log p_\theta(x) = E_{\theta_J}(x_J) + \sum_{j=1}^{J} \Big(\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j) + \log \bar{Z}_{\bar{\theta}_j}(x_j) \Big) + \text{cst}, \tag{A.18}$$

where $\bar{Z}_{\bar{\theta}_j}(x_j)$ is the normalization constant for $\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j)$. To retrieve the global negative log-likelihood $-\log p_\theta(x)$, we thus compute an approximation of $-\log \bar{Z}_{\bar{\theta}_j}(x_j)$ with a parametric family $F_{\tilde{\theta}_j}$.

The parameters $\tilde{\theta}_j$ of the approximation of the normalizing factors $\bar{Z}_{\bar{\theta}_j}$ can be learned in a manner similar to denoising score matching. Indeed, using the identity

$$-\nabla_{x_j} \log \bar{Z}_{\bar{\theta}_j}(x_j) = \mathbb{E}\Big[\nabla_{x_j} \bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j) \,|\, x_j \Big],$$

which can be proven by a direct computation of the gradient, the parameters $\tilde{\theta}_j$ can be estimated by minimizing

$$\tilde{\ell}_j(\tilde{\theta}_j) = \mathbb{E}\left[\left\| \nabla_{x_j} F_{\tilde{\theta}_j} - \nabla_{x_j} \bar{E}_{\bar{\theta}_j} \right\|^2 \right]. \tag{A.19}$$

For an exponential model $F_{\tilde{\theta}_j} = \tilde{\theta}_j^{\mathrm{T}} \tilde{\Phi}_j$ with a fixed potential vector $\tilde{\Phi}_j$, eq. (A.19) is quadratic in $\tilde{\theta}$ and admits a closed-form solution:

$$\tilde{\theta}_j = \mathbb{E}\Big[\nabla_{x_j} \tilde{\Phi}_j \nabla_{x_j} \tilde{\Phi}_j^{\mathrm{T}} \Big]^{-1} \mathbb{E}\Big[\nabla_{x_j} \tilde{\Phi}_j \nabla_{x_j} \bar{E}_{\bar{\theta}_j} \Big].$$

We finally obtain the energy decomposition

$$-\log p_\theta(x) = E_{\theta_J}(x_J) + \sum_{j=1}^{J} \Big(\bar{E}_{\bar{\theta}_j}(x_j, \bar{x}_j) - F_{\tilde{\theta}_j}(x_j) \Big) + \text{cst}. \tag{A.20}$$

This score-based method is much faster and simpler to implement than likelihood-based methods such as the thermodynamic integration of Marchand et al. (2022), which requires generation of many samples while varying the parameters $\bar{\theta}_j$ of the conditional energy $\bar{E}_{\bar{\theta}_j}$.

A.4.2 Parameterized free-energy models

The potential vector $\tilde{\Phi}_j$ is modeled in the class of eq. (2.11), following Marchand et al. (2022) and similarly to Appendix A.2.1:

$$F_{\tilde{\theta}_j}(x_j) = \frac{1}{2} x_j^{\mathrm{T}} \tilde{K}_j x_j + \tilde{V}_j(x_j) + \sum_i \tilde{v}_j(x_j[i])$$
$$\tilde{v}_j(t) = \sum_k \tilde{\alpha}_{j,k} \tilde{\rho}_{j,k}(t),$$

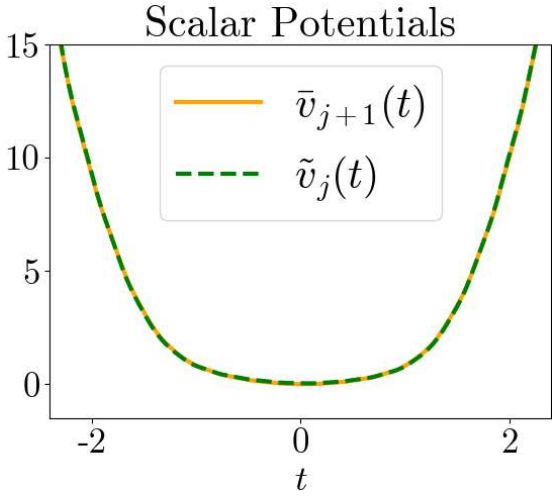

FIGURE A.5: For φ^4 at β_c, the conditional potentials $\bar{v}_{j+1}$ and free-energy potential $\tilde{v}_j$ cancel out. Only $j = 1$ is shown, other scales show similar behavior.

which gives $\tilde{\theta}_j = (\tilde{K}_j, \tilde{\alpha}_{j,k})_k$ and an associated potential vector

$$\tilde{\Phi}_j(x_j) = \left(\frac{1}{2} x_j x_j^{\mathrm{T}}, \tilde{\rho}_{j,k}(x_j)\right)_k.$$

A.4.3 Multiscale energy decomposition

We now expand the models for the conditional energies $\bar{E}_{\bar{\theta}_j}$ and the so-called free energies $F_{\tilde{\theta}_j}$ in eq. (A.20). All the quadratic terms $(K_J, \bar{K}_j, \tilde{K}_j)_j$ can be regrouped in an equivalent quadratic term K. We then have

$$\begin{aligned} -\log p_\theta(x) &= \frac{1}{2} x^{\mathrm{T}} K x + \sum_i \left[v_J(x_J[i]) + \sum_{j=1}^{J} (\bar{v}_j(x_{j-1}[i]) - \tilde{v}_j(x_j[i])) \right] \\ &= \frac{1}{2} x^{\mathrm{T}} K x + \sum_i \left[\bar{v}_1(x_0[i]) + \sum_{j=1}^{J} (\bar{v}_{j+1}(x_j[i]) - \tilde{v}_j(x_j[i])) \right], \end{aligned}$$

with $\bar{v}_{J+1} = v_J$. This defines multiscale scalar potentials V_j:

$$\begin{aligned} V_j &= \bar{v}_{j+1} - \tilde{v}_j, \\ V_0 &= \bar{v}_1, \end{aligned}$$

such that we have the global negative log-likelihood or energy function:

$$-\log p_\theta(x) = \frac{1}{2} x^{\mathrm{T}} K x + \sum_{j=0}^{J} \sum_i V_j(x_j[i]).$$

For φ^4 at critical temperature, as derived by Marchand et al. (2022), the only non-zero scalar potential will be V_0. The other V_j potentials are zero, up to a quadratic term.

As a numerical test, Figure A.5 verifies that on φ^4 at critical temperature, $\bar{v}_{j+1}$ and $\tilde{v}_j$ indeed cancel out so that $V_j = 0$ for $j > 0$. In order to ensure that the quadratic difference mentioned above vanishes, we subtract to $\tilde{v}_j$ the quadratic interpolation of $\tilde{v}_j - \bar{v}_{j+1}$.

A.5 Proof of Proposition 2.3

We directly compute the Hessian

$$\begin{aligned} -\nabla^2_{\bar{x}_1} \log p(\bar{x}_1|x_1) &= -\bar{G}_1 \nabla^2_x \log p(x) \bar{G}_1^{\mathrm{T}} \\ &= \bar{G}_1 \Big(K - \operatorname{diag}\Big((v''(x[i])) \Big)_i \Big) \bar{G}_1^{\mathrm{T}}, \end{aligned}$$

where we have used

$$p(\bar{x}_1|x_1) = \frac{p(x)}{p(x_1)}.$$

Both terms in the Hessian can now be bounded from below. The assumption on the range of $\bar{G}_1$ implies that

$$\bar{G}_1 K \bar{G}_1^{\mathrm{T}} \succeq \lambda |\omega_0|^\eta \operatorname{Id},$$

and the assumption on v'' implies that

$$\bar{G}_1 \operatorname{diag}\Big((v''(x[i])) \Big)_i \bar{G}_1^{\mathrm{T}} \succeq -\gamma \bar{G}_1 \bar{G}_1^{\mathrm{T}} = -\gamma \operatorname{Id},$$

where we have used the fact that $\bar{G}_1$ is an orthogonal projector.

Combining the two then gives

$$-\nabla^2_{\bar{x}_1} \log p(\bar{x}_1|x_1) \succeq (\lambda |\omega_0|^\eta - \gamma) \operatorname{Id},$$

and the assumption on $|\omega_0|$ guarantees that $\lambda |\omega_0|^\eta - \gamma > 0$. Similarly, the assumption $v'' \leq \delta$ implies that

$$-\nabla^2_{\bar{x}_1} \log p(\bar{x}_1|x_1) \preceq (\lambda \Omega^\eta + \delta) \operatorname{Id},$$

where $\Omega = \sup |\omega|$ is the maximum frequency, which concludes the proof.

Appendix

B Appendix for Chapter 3

Chapter content

B.1 WSGM algorithm 145
B.2 Introduction to the fast orthogonal wavelet transform 146
B.3 Experimental details on Gaussian experiments 147
B.4 Experimental details on the φ^4 model 148
B.5 Experimental details on CelebA-HQ 148

B.1 WSGM algorithm

In Algorithm B.1, we provide the pseudocode for WSGM. Notice that the training of score models at each scale can be done in parallel, while the sampling is done sequentially one scale after the next.

Algorithm B.1 Wavelet Score-based Generative Model

Require: J, N_{iter}, N, T,$\{\bar{\theta}_{j,0}, \theta_{J,0}\}_{j=0}^{J}$, $\{x_0^m\}_{m=1}^{M}$
/// WAVELET TRANSFORM ///
for $j \in \{1, \dots, J\}$ **do**
  for $m \in \{1, \dots, M\}$ **do**
    $x_j^m = \gamma_j^{-1} G x_{j-1}^m$, $\bar{x}_j^m = \gamma_j^{-1} \bar{G} x_{j-1}^m$ ▷ Wavelet transform of the dataset
  end for
end for
/// TRAINING ///
Train score network $s_{\theta_J^\star}$ at scale J with dataset $\{x_J^m\}_{m=0}^{M}$ ▷ Unconditional SGM training
for $j \in \{J, \dots, 1\}$ **do** ▷ Can be run in parallel
  for $n \in \{0, \dots, N_{\text{iter}} - 1\}$ **do**
    Sample $(\bar{x}_{j,0}, x_j)$ from $\{\bar{x}_j^m, x_j^m\}_{m=1}^{M}$
    Sample t in $[0, T]$ and $\bar{Z} \sim \mathrm{N}(0, \mathrm{Id})$
    $\bar{x}_{j,t} = \mathrm{e}^{-t} \bar{x}_{j,0} + (1 - \mathrm{e}^{-2t})^{1/2} \bar{Z}$
    $\ell(\bar{\theta}_{j,n}) = \| (\mathrm{e}^{-t} \bar{x}_{j,0} - \bar{x}_{j,t}) - (1 - \mathrm{e}^{-2t})^{1/2} \bar{s}_{\bar{\theta}_{j,n}}(t, \bar{x}_{j,t} | x_j) \|^2$
    $\bar{\theta}_{j,n+1} = \texttt{optimizer\_update}(\bar{\theta}_{j,n}, \ell(\bar{\theta}_{j,n}))$ ▷ ADAM optimizer step
  end for
  $\bar{\theta}_j^\star = \bar{\theta}_{j,N_{\text{iter}}}$
end for
/// SAMPLING ///
$x_J = \texttt{EulerMaruyama}(T, N, s_{\theta_J^\star})$ ▷ Euler-Maruyama recursion following (3.19)
for $j \in \{J, \dots, 1\}$ **do**
  $\bar{x}_j = \texttt{EulerMaruyama}(T, N, \bar{s}_{\bar{\theta}_j^\star}(\cdot, \cdot | x_j))$ ▷ Euler-Maruyama recursion following (3.20)
  $x_{j-1} = \gamma_j G^{\mathrm{T}} x_j + \gamma_j \bar{G}^{\mathrm{T}} \bar{x}_j$ ▷ Wavelet reconstruction
end for
return $\{\bar{\theta}_j^\star, \theta_J^\star\}_{j=1}^{J}$, x_0 ▷ Returns learned parameters and generated samples

B.2 Introduction to the fast orthogonal wavelet transform

This section introduces the fast orthogonal wavelet transform introduced in Mallat (1989). It is computed with convolutional operators G and $\bar{G}$. In this section, we deal with the non-normalized wavelet transform, which is obtained by setting $\gamma_j = 1$. To avoid confusion with normalized wavelet coefficients $(x_j, \bar{x}_j)$, we denote the non-normalized wavelet coefficients with a w exponent: $(x_j^w, \bar{x}_j^w)$.

Let x_0^w be a signal. The index u in $x_0^w(u)$ belongs to an n-dimensional grid of linear size L and hence with L^n sites, with $n = 2$ for images. Let us denote x_j^w the coarse-grained version of x_0^w at a scale 2^j defined over a coarser grid with intervals 2^j and hence $(2^{-j}L)^n$ sites. The coarser signal x_j^w is iteratively computed from x_{j-1}^w by applying a coarse-graining operator, which acts as a scaling filter G which eliminates high frequencies and subsamples the grid

$$(Gx_{j-1}^w)(u) = \sum_{u'} x_{j-1}^w(u')\, G(2u - u') \ . \tag{B.1}$$

The index u on the left-hand side runs on the coarser grid, whereas u' runs on the finer one.

The degrees of freedom of x_{j-1}^w that are not in x_j^w are encoded in orthogonal wavelet coefficients $\bar{x}_j^w$. The representation $(x_j^w, \bar{x}_j^w)$ is an orthogonal change of basis calculated from x_{j-1}^w. The coarse signal x_j^w is calculated in (B.1) with a low-pass scaling filter G and a subsampling. In dimension n, the wavelet coefficients $\bar{x}_j^w$ have $2^n - 1$ channels computed with a convolution and subsampling operator $\bar{G}$. We thus have

$$x_j^w = G\, x_{j-1}^w \quad \text{and} \quad \bar{x}_j^w = \bar{G}\, x_{j-1}^w. \tag{B.2}$$

The wavelet filter $\bar{G}$ computes $2^n - 1$ wavelet coefficients $\bar{x}_j^w(u, k)$ indexed by $1 \le k \le 2^n - 1$, with separable high-pass filters $\bar{G}_k(u)$

$$\bar{x}_j^w(u, k) = \sum_{u'} x_{j-1}^w(u')\, \bar{G}_k(2u - u').$$

As an example, the Haar wavelet leads to a block averaging filter G. In dimension $n = 1$

$$x_j^w(u) = \frac{x_{j-1}^w(2u) + x_{j-1}^w(2u+1)}{\sqrt{2}},$$

and there is a single wavelet channel in $\bar{x}_j^w$. The corresponding wavelet filter $\bar{G}$ computes the wavelet coefficients with increments divided by $\sqrt{2}$

$$\bar{x}_j^w(u) = \frac{x_{j-1}^w(2u) - x_{j-1}^w(2u+1)}{\sqrt{2}}. \tag{B.3}$$

If $n = 2$, then there are $2^n - 1 = 3$ wavelet channels as shown in Figure 3.1.

The fast wavelet transform cascades (B.2) for $1 \le j \le J$ to compute the decomposition of the high-resolution signal x_0^w into its orthogonal wavelet representation over J scales

$$\{x_J^w\ ,\ \bar{x}_j^w\}_{1 \le j \le J}. \tag{B.4}$$

The wavelet orthonormal filters G and $\bar{G}$ define a unitary transformation, satisfying

$$\bar{G}G^{\mathrm{T}} = G\bar{G}^{\mathrm{T}} = 0 \quad \text{and} \quad G^{\mathrm{T}}G + \bar{G}^T\bar{G} = \mathrm{Id}\ , \tag{B.5}$$

where Id is the identity. Conjugate mirror conditions are given in Mallat (1989) on the Fourier transforms of G and $\bar{G}$ to build such unitary filters. The filtering equations (B.2) can then be inverted with the adjoint operators

$$x_{j-1}^w = G^{\mathrm{T}} x_j^w + \bar{G}^{\mathrm{T}} \bar{x}_j^w \ . \tag{B.6}$$

The adjoint G^{T} enlarge the grid size of x_j^w by inserting a zero between each coefficients, and then filters the output:

$$(G^{\mathrm{T}} x_j^w)(u) = \sum_{u'} x_j^w(u')\, G(2u' - u).$$

The adjoint of $\bar{G}$ performs the same operations over the $2^n - 1$ channels and adds them:

$$(\bar{G}^{\mathrm{T}} \bar{x}_j^w)(u) = \sum_{k=1}^{2^n-1} \sum_{u'} \bar{x}_j^w(u', k)\, \bar{G}_k(2u' - u).$$

The fast inverse wavelet transform Mallat (1989) recovers x_0^w from its wavelet representation (B.4) by progressively recovering x_{j-1}^w from x_j^w and $\bar{x}_j^w$ with (B.6), for j going from J to 1.

B.3 Experimental details on Gaussian experiments

We now give some details on the experiments in Section 3.3.2 (Figure 3.2). We use the exact formulas for the Stein score of p_t in this case: if $x_0 \sim \mathcal{N}(M, \Sigma)$, then $x_t \sim N(M_t, \Sigma_t)$ with $M_t = e^{-t} M$ and

$$\Sigma_t = \mathrm{e}^{-2t}\Sigma + (1 - \mathrm{e}^{-2t})\,\mathrm{Id}\,.$$

Under an ideal situation where there is no score error, the discretization of the (backward) generative process is given by equation

$$x_{k+1} = ((1+\delta)\,\mathrm{Id} - 2\delta\Sigma_{T-k\delta}^{-1})x_k + 2\delta\Sigma_{T-k\delta}^{-1} M_{T-k\delta} + \sqrt{2\delta} z_{k+1}\,, \tag{B.7}$$

where δ is the uniform step size and z_k are i.i.d. white Gaussian random variables. For the SGM case, $M = 0$. The starting step of this discretization is itself $x_0 \sim \mathcal{N}(0, \mathrm{Id})$. From this formula, the covariance matrix $\hat{\Sigma}_k$ of x_k satisfies the recursion

$$\hat{\Sigma}_{k+1} = ((1+\delta)\,\mathrm{Id} - 2\delta\Sigma_{T-k\delta}^{-1})\hat{\Sigma}_k((1+\delta)\,\mathrm{Id} - 2\delta\Sigma_{T-k\delta}^{-1}) + 2\delta\,\mathrm{Id}\,, \tag{B.8}$$

from which we can exactly compute $\hat{\Sigma}_k$ for very k, and especially for $k = N = T/\delta$, as a function of Σ, the final time T, and the step size δ. In all our experiments, we choose stationary processes: their covariance Σ is diagonal in a Fourier basis, with eigenvalues (*power spectrum*) noted $\hat{P}_k$. All the x_k remain stationary so $\hat{\Sigma}_k$ is still diagonal in a Fourier basis, with power spectrum noted $\hat{P}_k$. The error displayed in the left panel of Figure 3.2 is

$$\|\hat{P}_N - P\| = \max_\omega |\hat{P}_N(\omega) - P(\omega)| / \max_\omega |P(\omega)|,$$

normalized by the operator norm of Σ.

The illustration in the middle panel of Figure 3.2, for WSGM, is done for simplicity only at one scale (ie, at $j = 1$ in Algorithm B.1): instead of stacking the full cascade of conditional distributions for all $j = J, \dots, 1$, we use the true low-frequencies $x_{j,0} = x_1$. Here, we use Daubechies-4 wavelets. We sample $\bar{x}_{j,0}$ using the Euler-Maruyama recursion (B.7)-(B.8) for the conditional distribution. We recall that in the Gaussian case, $\bar{x}_1$ and x_1 are jointly Gaussian. The conditional distribution of $\bar{x}_1$ given x_1 is known to be $\mathcal{N}(Ax_1, \Gamma)$, where

$$A = -\mathrm{Cov}(\bar{x}_1, x_1)\mathrm{Var}(x_1)^{-1}, \qquad \Gamma = \mathrm{Var}(\bar{x}_1) - \mathrm{Cov}(\bar{x}_1, x_1)\mathrm{Var}(x_1)^{-1}\mathrm{Cov}(\bar{x}_1, x_1)^{\mathrm{T}}. \tag{B.9}$$

We solve the recursion (B.8) with a step size δ and $N = T/\delta$ steps; the sampled conditional wavelet coefficients $\bar{x}_{j,0}$ have conditional distribution noted $\mathcal{N}(\hat{A}_N x, \hat{\Gamma}_N)$. The full covariance of $(\tilde{x}_{j,0}, \bar{x}_{j,0})$, written in the basis given by the high/low frequencies, is now given by

$$\hat{\Sigma}_N = \begin{bmatrix} \hat{\Gamma}_N & \mathrm{Cov}(x_1, \bar{x}_1)\hat{A}_N^{\mathrm{T}} \\ \hat{A}_N \mathrm{Cov}(x_1, \bar{x}_1)^{\mathrm{T}} & \mathrm{Cov}(x_1, x_1) \end{bmatrix}.$$

Figure 3.2, middle panel compares the eigenvalues (power spectrum) of these covariances, as a function of δ, with the ones of Σ.

The right panel of Figure 3.2 gives the smallest N needed to reach $\|\hat{P}_N - P\| = 0.1$ in both cases (SGM and WSGM), based on a power law extrapolation of the curves $N \mapsto \hat{P}_N$.

B.4 Experimental details on the φ^4 model

Training data and wavelets. We used samples from the φ^4 model generated using a classical MCMC algorithm — the sampling script will be publicly available in our repository.

The wavelet decompositions of our fields were performed using Python's `pywavelets` package and `Pytorch Wavelets` package. For synthetic experiments, we used the Daubechies wavelets with $p = 4$ vanishing moments (see Mallat, 2008, Section 7.2.3).

Score model. At the first scale $j = 0$, the distribution of the φ^4 model falls into the general form given in (3.23), and it is assumed that at each scale j, the distribution of the field at scale j still assumes this shape — with modified constants and coupling parameters. The score model we use at each scale is given by

$$s_{K,\theta}(x) = \tfrac{1}{2} x^{\mathrm{T}} K x + \textstyle\sum_u (\theta_1 v_1(x(u)) + \cdots + \theta_m v_m(x(u))), \tag{B.10}$$

where the parameters are $K, \theta_1, \ldots, \theta_m$ and v_i are a family of smooth functions. One can also represent this score as $s_{K,\theta} = K \cdot x x^{\mathrm{T}} + \theta^{\mathrm{T}} U(x)$ where $U_i(x) = \sum_u v_i(x(u))$.

Learning. We trained our various algorithms using SGM or WSGM up to a time $T = 5$ with $n_{\text{train}} = 2000$ steps of forward diffusion. At each step t, the parameters were learned by minimizing the score loss

$$\ell(K, \theta) = \mathbb{E}[|\nabla s_{K,\theta}(x_t)|^2 + 2\Delta_x s_{K,\theta}(x_t)]$$

using the Adam optimiser with learning rate `lr` $= 0.01$ and default parameters α, β. At the start of the diffusion ($t = 0$) we use 10000 steps of gradient descent. For $t > 1$, we use only 100 steps of gradient descent, but initialized at (K_{t-1}, θ_{t-1}).

Sampling. For the sampling, we used uniform steps of discretization.

For the error metric, we first measure the L^2-norm between the power spectra $P, \hat{P}$ of the true φ^4 samples and our synthesized examples; more precisely, we set

$$D_1 = \|P - \hat{P}\|^2.$$

This error on second-order statistics is perfectly suitable for Gaussian processes, but must be refined for non-Gaussian processes. We also consider the total variation distance between the histograms of the marginal distributions (in the case of two-dimensions, pixel-wise histograms). We note this error D_2; our final error measure is $D_1 + D_2$.

B.5 Experimental details on CelebA-HQ

Data. We use Haar wavelets. The 128×128 original images are thus successively brought to the 64×64 and 32×32 resolutions, separately for each color channel. Each of the 3 channels of x_j and 9 channels of $\bar{x}_j$ are normalized to have zero mean and unit variance.

Architecture. Following Nichol and Dhariwal (2021), both the conditional and unconditional scores are parameterized by a neural network with a U-Net architecture. It has 3 residual blocks at each scale, with a base number of channels of $C = 128$. The number of channels at the k-th scale is $a_k C$, where the multipliers $(a_k)_k$ depend on the resolution of the generated images. These multipliers are $(1, 2, 2, 4, 4)$ for models at the 128×128 resolution, $(2, 2, 4, 4)$ for models at the 64×64 resolution, $(4, 4)$ for the conditional model at the 32×32 resolution, and $(1, 2, 2, 2)$ for the unconditional model at the 32×32 resolution. All models include multi-head attention layers in blocks operating on images at resolutions 16×16 and 8×8. The conditioning on the low frequencies x_j is done with a simple input concatenation along channels, while conditioning on time is done through affine rescalings with learned time embeddings at each GroupNorm layer (Nichol and Dhariwal, 2021; Saharia et al., 2021).

Training. The networks are trained with the (conditional) denoising score matching losses

$$\ell(\theta_J) = \mathbb{E}_{x_J,t,z}\left[||s_{\theta_J}(t, \mathrm{e}^{-t}x_J + \sqrt{1-\mathrm{e}^{-2t}}z) - \frac{z}{\sqrt{1-\mathrm{e}^{-2t}}}||^2\right], \tag{B.11}$$

$$\ell(\bar{\theta}_j) = \mathbb{E}_{\bar{x}_j,x_j,t,z}\left[||\bar{s}_{\bar{\theta}_j}(t, \mathrm{e}^{-t}\bar{x}_j + \sqrt{1-\mathrm{e}^{-2t}}z \,|\, x_j) - \frac{z}{\sqrt{1-\mathrm{e}^{-2t}}}||^2\right], \tag{B.12}$$

where $z \sim \mathcal{N}(0, \mathrm{Id})$ and the time t is distributed as Tu^2 with $u \sim \mathcal{U}([0, 1])$. We fix the maximum time $T = 5$ for all scales. Networks are trained for 5×10^5 gradient steps with a batch size of 128 at the 32×32 resolution and 64 otherwise. We use the Adam Kingma and Ba (2014) optimizer with a learning rate of 10^{-4} and no weight decay.

Sampling. For sampling, we use model parameters from an exponential moving average with a rate of 0.9999. For each number of discretization steps N, we use the Euler-Maruyama discretization with a uniform step size $\delta_k = T/N$ starting from $T = 5$. This discretization scheme is used at all scales. For FID computations, we generate $30,000$ samples in each setting.

Appendix

C Appendix for Chapter 4

Chapter content

C.1 Proof of Theorem 4.1 151
C.2 Proof of equation (4.5) 152
C.3 Training and architecture details 152
C.4 Wavelet conditional synthesis algorithm 153

C.1 Proof of Theorem 4.1

To simplify notation, we drop the j subscript. Let I (resp. J) denote the set of indices of pixel values of $\bar{x}$ (resp. x). If S is a set of indices, we denote $\bar{x}(S) = (\bar{x}(i))_{i \in S \cap I}$. Let G be a graph whose nodes are $I \cup J$. For each $i \in I$, let $N(i) \subseteq I \cup J$ be the neighborhood of node i, with $i \notin N(i)$, and $N_+(i) = N(i) \cup \{i\}$.

To prove Theorem 4.1, we need to show that the local Markov property

$$\forall i \in I,\ p(\bar{x}(i) \,|\, \bar{x}(I \setminus \{i\}), x) = p(\bar{x}(i) \,|\, \bar{x}(N(i)), x(N_+(i))), \tag{C.1}$$

is equivalent to the conditional score being computable with RFs restricted to neighborhoods

$$\forall i \in I,\ \frac{\partial \log p}{\partial \bar{x}(i)}(\bar{x} \,|\, x) = f_i(\bar{x}(N_+(i)), x(N_+(i))), \tag{C.2}$$

for some functions f_i.

We first prove that eq. (C.1) implies eq. (C.2). Let $i \in I$. We have the following factorization of the probability distribution:

$$\begin{aligned} p(\bar{x} \,|\, x) &= p(\bar{x}(i) \,|\, \bar{x}(I \setminus \{i\}), x)\, p(\bar{x}(I \setminus \{i\}) \,|\, x) \\ &= p(\bar{x}(i) \,|\, \bar{x}(N(i)), x(N_+(i)))\, p(\bar{x}(I \setminus \{i\}) \,|\, x), \end{aligned}$$

where we have used eq. (C.1) in the last step. Then, taking the logarithm and differentiating, only the first term remains:

$$\frac{\partial \log p}{\partial \bar{x}(i)}(\bar{x} \,|\, x) = \frac{\partial \log p}{\partial \bar{x}(i)}(\bar{x}(i) \,|\, \bar{x}(N(i)), x(N_+(i))),$$

which proves eq. (C.2).

Reciprocally, we now prove that eq. (C.2) implies eq. (C.1). Let $i \in I$, and $\delta_i(j) = \delta_{ij}$ where δ_{ij} is the Kronecker delta. We have, by integrating the partial derivative,

$$\begin{aligned} \log p(\bar{x} \,|\, x) &= \log p(\bar{x} - \bar{x}(i)\delta_i \,|\, x) - \int_0^1 \frac{\partial \log p}{\partial \bar{x}(i)}(\bar{x} - t\bar{x}(i)\delta_i \,|\, x) \mathrm{d}t \\ &= \log p(\bar{x} - \bar{x}(i)\delta_i \,|\, x) - \int_0^1 f_i(\bar{x}(N_+(i)) - t\bar{x}(i)\delta_i, x(N_+(i))) \mathrm{d}t, \end{aligned}$$

where we have used eq. (C.2) in the last step. Note that the first term does not depend on $\bar{x}(i)$, while the second term only depends on $\bar{x}(N_+(i))$ and $x(N_+(i))$. This implies that when we condition on $\bar{x}(N(i))$ and $x(N_+(i))$, the density factorizes as a term which only involves $\bar{x}(i)$ and a term which does not involve $\bar{x}(i)$. This further implies conditional independence and thus eq. (C.1).

C.2 Proof of equation (4.5)

Miyasawa's remarkable result (Miyasawa, 1961), sometimes attributed to Tweedie (as communicated by Robbins, 1956), is simple to prove (Raphan and Simoncelli, 2007). The observation distribution, $p(y)$ is obtained by marginalizing $p(y, x)$:

$$p(y) = \int p(y|x)p(x)\mathrm{d}x = \int g(y-x)p(x)\mathrm{d}x,$$

where the noise distribution $g(z)$ is Gaussian. The gradient of the observation density is then

$$\nabla_y\, p(y) = \frac{1}{\sigma^2}\int (x-y)g(y-x)p(x)\mathrm{d}x = \frac{1}{\sigma^2}\int (x-y)p(y,x)\mathrm{d}x.$$

Multiplying both sides by $\sigma^2/p(y)$ and separating the right side into two terms gives

$$\sigma^2\frac{\nabla_y\, p(y)}{p(y)} = \int xp(x|y)\mathrm{d}x - \int yp(x|y)\mathrm{d}x = \hat{x}(y) - y.$$

C.3 Training and architecture details

Architecture. The terminal low-pass CNN and all cCNNs are "bias-free": we remove all additive constants from convolution and batch-normalization operations (i.e., the batch normalization does not subtract the mean) (Mohan et al., 2019). All networks contain 21 convolutional layers with no subsampling, each consisting of 64 channels. Each layer, except for the first and the last, is followed by a ReLU non-linearity and bias-free batch-normalization. Thus, the transformation is both homogeneous (of order 1) and translation-invariant (apart from handling of boundaries), at each scale. All convolutional kernels in the low-pass CNN are of size 3×3, resulting in a 43×43 RF size and $665,856$ parameters in total. Convolutional kernels in the cCNNs are adjusted to achieve different RF sizes. For example, a 13×13 RF arises from choosing 3×3 kernels in every 4^{th} layer and 1×1 (i.e., pointwise linear combinations across all channels) for the rest, resulting in a total of $214,144$ parameters. For comparison, we also trained conventional (non-multiscale) CNNs for denoising. For RF 43×43, we used the same architecture as for the coarsest scale band of the multiscale denoiser: 21 bias-free convolutional layers with no subsampling. To create smaller RFs, we followed the same strategy of setting some filter sizes in the intermediate layer to 1×1.

Training. For experiments shown in Figures 4.3 and 4.4, we use $202,499$ training and 100 test images of resolution 160×160 from the CelebA dataset (Liu et al., 2015). For experiments shown in Figures 4.5 and 4.6, we use $29,900$ train and 100 test images, drawn from the CelebA HQ dataset (Karras et al., 2018) at 320×320 resolution. We follow the training procedure described in (Mohan et al., 2019), minimizing the mean squared error in denoisingd images corrupted by i.i.d. Gaussian noise with standard deviations drawn from the range [0, 1] (relative to image intensity range [0, 1]). Training is carried out on batches of size 512. Note that all denoisers are universal and blind: they are trained to handle a range of noise, and the noise level is not provided as input to the denoiser. These properties are exploited by the sampling algorithm, which can operate without manual specification of the step size schedule Kadkhodaie and Simoncelli (2021).

C.4 Wavelet conditional synthesis algorithm

Sampling from both the CNN and cCNN denoisers is achieved using a slightly modified version of the algorithm of Kadkhodaie and Simoncelli (2021), as defined in Algorithm C.1. This method uses only two hyperparameters, aside from initial and final noise levels, and their settings are more forgiving than those of backward SDE discretization parameters in score-based diffusions. A step size parameter, $h \in [0, 1]$, controls the trade-off between computational efficiency and visual quality. A stochasticity parameter, $\beta \in (0, 1]$, controls the amount of noise injected during the gradient ascent. For the examples in Figures 4.5 and 4.6, we chose $h = 0.01$, $\sigma_0 = 1$, $\beta = 0.1$ and $\sigma_\infty = 0.01$.

Image synthesis is initialized with a terminal low-pass image (either sampled from the associated CNN, or computed from a test image), and successively sampling from the wavelet conditional distributions at each scale, as defined in Algorithm C.2.

Algorithm C.1 Sampling via ascent of the log-likelihood gradient from a denoiser residual

Require: denoiser f, step size h, initial noise level σ_0, final noise level σ_∞

$t = 0$
Draw $x_0 \sim \mathcal{N}(0, \sigma_0^2 \mathrm{Id})$
while $\sigma_t \geq \sigma_\infty$ **do**
    $t \leftarrow t + 1$
    $d_t \leftarrow f(x_{t-1}) - x_{t-1}$ ▷ Compute the score from the denoiser residual
    $\sigma_t^2 \leftarrow ||d_t||^2 / N$ ▷ Compute the current noise level for stopping criterion
    $\gamma_t^2 = \Big((1 - \beta h)^2 - (1 - h)^2\Big)\sigma_t^2$
    Draw $z_t \sim \mathcal{N}(0, I)$
    $x_t \leftarrow x_{t-1} + h d_t + \gamma_t z_t$ ▷ Perform a partial denoiser step to remove a fraction of the noise
end while
return x_t

Algorithm C.2 Wavelet Conditional Synthesis

Require: number of scales J, low-pass image x_J, conditional denoisers $(f_j)_{1 \leq j \leq J}$, step size h, initial noise level σ_0, final noise level σ_∞

for $j \in \{J, \ldots, 1\}$ **do**
    $\bar{x}_j \leftarrow \mathrm{DrawSample}(f_j(\cdot, x_j), h, \sigma_0, \sigma_\infty)$ ▷ Wavelet conditional sampling
    $x_{j-1} \leftarrow W^T(\bar{x}_j, x_j)$ ▷ Wavelet reconstruction
end for
return x_0

Appendix

D Appendix for Chapter 5

Chapter content

D.1 Proof of Proposition 5.1 155
D.2 Proof of Theorem 5.2 156
D.3 Implementation and network dimensions 156

D.1 Proof of Proposition 5.1

We first prove the following lemma:

Lemma D.1. *If Φ is linear, then the Fisher ratio is decreased (or equal) and the optimal linear classification error is increased (or equal).*

If Φ is linear, then it is a matrix $\in \mathbb{R}^{p\times d}$. We assume that Φ has rank p (and thus $p \leq d$) for the sake of simplicity. By applying a polar decomposition on $\Phi\Sigma_W^{1/2}$, we can write

$$\Phi = UP\Sigma_W^{-1/2},$$

where $U \in \mathbb{R}^{p\times p}$ is symmetric positive-definite and $P \in \mathbb{R}^{p\times d}$ verifies $PP^T = \mathrm{Id}$. The within-class covariance and class means of Φx are given by

$$\overline{\Sigma}_W = \Phi\Sigma_W\Phi^T = U^2,$$
$$\overline{\mu}_c = \Phi\mu_c = UP\Sigma_W^{-1/2}\mu_c.$$

The Fisher ratio of Φx is thus

$$\begin{aligned} C^{-1}\operatorname{Tr}(\overline{\Sigma}_W^{-1}\overline{\Sigma}_B) &= \operatorname*{Ave}_c \|\overline{\Sigma}_W^{-1/2}\bar{\mu}_c\|^2 \\ &= \operatorname*{Ave}_c \|P\Sigma_W^{-1/2}\mu_c\|^2 \\ &\leq \operatorname*{Ave}_c \|\Sigma_W^{-1/2}\mu_c\|^2 \\ &= C^{-1}\operatorname{Tr}(\Sigma_W^{-1}\Sigma_B), \end{aligned}$$

so Φ decreases the Fisher ratio. Besides, if (W, b) is the optimal linear classifier on Φx, then $(W\Phi, b)$ is a linear classifier on x, and thus has a larger (or equal) error than the optimal linear classifier on x.

Now, if Φ has a linear inverse Φ^{-1}, we apply the Lemma D.1 to $x' = \Phi x$ and $\Phi' = \Phi^{-1}$ (so that $\Phi' x' = x$), which concludes the proof.

Additionally, we can see from the proof of the lemma that a linear Φ preserves the Fisher ratio if and only if $\|P\Sigma_W^{-1/2}\mu_c\| = \|\Sigma_W^{-1/2}\mu_c\|$ for all c. This happens when $\Sigma_W^{-1/2}\mu_c$ is in the orthogonal of $\operatorname{Ker} P = \operatorname{Ker} UP = \operatorname{Ker}\Phi\Sigma_W^{1/2}$, which means that $\Sigma_W^{-1}\mu_c$ is in the orthogonal of $\operatorname{Ker}\Phi$. When Φ is an orthogonal projector, the orthogonal of $\operatorname{Ker}\Phi$ is the range of Φ.

D.2 Proof of Theorem 5.2

We choose $x = ru$ with $u \sim \mathcal{U}(\mathbb{S}^{d-1})$ and $r \in]0,1]$ to be determined, with r and u independent. Let us fix $p \geq d$, $D \in \mathbb{R}^{d\times p}$, $\theta \in \mathbb{R}^p$ and $b \in \mathbb{R}$. With $g(x) = \theta^{\mathrm{T}} \rho_{rt} D^{\mathrm{T}} x + b$, we have

$$\begin{aligned} g(x) &= \sum_{m=1}^{p} w_m \rho_r(r\langle u, f_m\rangle - \lambda) + b \\ &= r \sum_{m=1}^{p} w_m \rho_r(\langle u, f_m\rangle - \lambda/r) + b\,. \end{aligned}$$

If $\lambda = 0$, this gives $g(x) = r\theta^{\mathrm{T}} \rho_r(D^{\mathrm{T}} u) + b$ which is an affine function of r. Therefore, its sign can change at most once. We choose $h(x) = \cos(2\pi\|x\|)$ so that

$$\mathrm{sgn}(h(x)) = \begin{cases} +1 & r < \frac{1}{4} \text{ or } \frac{3}{4} < r \\ -1 & \frac{1}{4} < r < \frac{3}{4} \end{cases}.$$

Now $g(x)$ is an affine function of r, so at least one of the following must occur:

$$\begin{cases} \mathrm{sgn}(g(x)) = -1 & r < \frac{1}{4} \\ \mathrm{sgn}(g(x)) = +1 & \frac{1}{4} < r < \frac{3}{4} \\ \mathrm{sgn}(g(x)) = -1 & \frac{3}{4} < r \end{cases}.$$

We finally choose $r \sim \mathcal{U}(0,1)$ and so we conclude that

$$\mathbb{P}[\mathrm{sgn}(g(x)) \neq \mathrm{sgn}(h(x))] \geq \frac{1}{4}\,.$$

If $\lambda > 0$, then when $r \leq \lambda$, we have $\langle u, f_m\rangle \leq \|u\|\|f_m\| \leq 1 \leq \lambda/r$, which means that $g(x) = b$ is constant. We thus choose $r \sim \mathcal{U}(0,\lambda)$, $h(x) = \cos(\pi/\lambda\|x\|)$ and so we conclude that

$$\mathbb{P}[\mathrm{sgn}(g(x)) \neq \mathrm{sgn}(h(x))] = \frac{1}{2} \geq \frac{1}{4}\,.$$

D.3 Implementation and network dimensions

All networks are trained with SGD with a momentum of 0.9 and a weight decay of 10^{-4} for the classifier weights, with no weight decay being applied to tight frames. The learning rate is set to 0.01 for all networks, with a Parseval regularization parameter $\alpha = 0.0005$. The batch size is 128 for all experiments. The scattering transform is based on the *Kymatio* package (Andreux et al., 2020). Standard data augmentation was used on CIFAR and ImageNet: horizontal flips and random crops for CIFAR, and random resized crops of size 224 and horizontal flips for ImageNet. Classification error on ImageNet validation set is computed on a single center-crop of size 224.

Non-linearity thresholds are set to $\lambda = 1.5\sqrt{d/p}$ for the soft-thresholding ρ_t, and $\lambda = \sqrt{d/p}$ for the thresholded rectifier ρ_{rt}. Here d and p represent the dimension of the patches the convolutional operators D and D^T act on. To ensure that the fixed threshold is well adapted to the scale of the input x, we normalize all its patches so that they have a norm of $\sqrt{d}$. For 1×1 convolutional operators as in S_C, this amounts to normalizing the channel vectors at each spatial location in x.

Two-layer networks. When learning a frame contraction directly on the input image, D^{T} is a convolutional operator over image patches of size $k \times k$ with a stride of $k/2$, where $k = 14$ for MNIST ($d = k^2 = 196$) and $k = 8$ for CIFAR ($d = 3k^2 = 192$). The frame D^{T} has p output channels, where $p = 2048$ for MNIST and $p = 8192$ for CIFAR. It thus maps each patch of dimension d to a channel vector of size $p \geq d$. Training lasts for 300 epochs, the learning rate being divided by 10 every 70 epochs.

	Φ	S_T	S_P	S_C	ResNet-18
ImageNet	Parameters	25.9M	27.6M	31.2M	11.7M

Table D.1: Number of parameters of scattering architectures on ImageNet. They are dominated by the size of the 1×1 orthogonal projectors P_j. Indeed, the wavelet tight frame W has a redundancy of $(L + 1/4)$, whereas in ResNet strided convolutions have a redundancy of $1/2$. This is due to the fact that W is not learned. However, W comes with a known structure across channels, which is beneficial for the analysis of the projectors P_j.

Scattering tree. We use $J = 3$ for MNIST and CIFAR and $J = 4$ for ImageNet. Each W uses $L = 8$ angles. It is followed by a standardization which sets the mean and variance of every channel to 0 and 1. We then learn a 1×1 convolutional orthogonal projector P_J to reduce the number of channels to $d = 512$. We finally apply a 1×1 spatial normalization, as before a tight frame thresholding. Training lasts for 300 epochs for MNIST and CIFAR (200 epochs for ImageNet), the learning rate being divided by 10 every 70 epochs (60 epochs for ImageNet).

Learned scattering. We use $J = 4$ for CIFAR and $J = 6$ for ImageNet. Each W uses $L = 8$ angles. Each P_j is an orthogonal projector which is a 1×1 convolution. It reduces the number of channels to d_j with $d_1 = 64$, $d_2 = 128$, $d_3 = 256$ and $d_4 = 512$. For ImageNet, we also have $d_5 = d_6 = 512$. It is followed by a normalization which sets the norm across channels of each spatial position to $\sqrt{d_j}$. D_j^{T} is a 1×1 convolutional tight frame with p_j output channels, where $p_1 = 1024$, $p_2 = 2048$, $p_3 = 4096$ and $p_4 = 8192$ for CIFAR, $p_1 = 512$, $p_2 = p_3 = 1024$ and $p_4 = p_5 = p_6 = 2048$ for ImageNet. Training lasts for 300 epochs for CIFAR (200 epochs for ImageNet), the learning rate being divided by 10 every 70 epochs (60 epochs for ImageNet).

Fisher ratios. Fisher ratios (eq. (5.1)) were computed using estimations of Σ_W and μ_c on the validation set. These estimations are unstable when the dimension d becomes large with respect to the number of data samples. To mitigate this, the Fisher ratios across layers from Table 5.3 were computed on the train set. Fisher ratios on ImageNet from Table 5.2 were computed only across channels, by considering each pixel as a distinct sample of the same class, in order to reduce dimensionality.

Appendix

E Appendix for Chapter 6

Chapter content

E.1 Proof of Theorem 6.1 159
E.2 Proof of equation (6.4) 159
E.3 Proof of Theorem 6.2 160
E.4 Proof of Theorem 6.3 161
E.5 Experimental details 162

E.1 Proof of Theorem 6.1

We have

$$\begin{aligned}\|x_\tau * \psi - e^{-i\xi\cdot\tau}(x*\psi)\|_\infty &= \|x*(\psi_\tau - e^{-i\xi\cdot\tau}\psi)\|_\infty && \text{by covariance of convolution,}\\ &\le \|\psi_\tau - e^{-i\xi\cdot\tau}\psi\|_2\|x\|_2 && \text{by Young's inequality,}\end{aligned}$$

and then

$$\begin{aligned}\|\psi_\tau - e^{-i\xi\cdot\tau}\psi\|_2^2 &= \frac{1}{(2\pi)^2}\int_{[-\pi,\pi]^2} |\widehat{\psi_\tau}(\omega) - e^{-i\xi\cdot\tau}\widehat{\psi}(\omega)|^2 \mathrm{d}\omega && \text{by Plancherel,}\\ &= \frac{1}{(2\pi)^2}\int_{[-\pi,\pi]^2} |e^{-i\omega\cdot\tau}\widehat{\psi}(\omega) - e^{-i\xi\cdot\tau}\widehat{\psi}(\omega)|^2 \mathrm{d}\omega && \text{since } \psi_\tau(u)=\psi(u-\tau),\\ &= \frac{1}{(2\pi)^2}\int_{[-\pi,\pi]^2} |e^{-i\omega\cdot\tau} - e^{-i\xi\cdot\tau}|^2|\widehat{\psi}(\omega)|^2 \mathrm{d}\omega\\ &\le \frac{1}{(2\pi)^2}\int_{[-\pi,\pi]^2} |(\omega-\xi)\cdot\tau|^2|\widehat{\psi}(\omega)|^2 \mathrm{d}\omega && \text{since } x\in\mathbb{R}\mapsto e^{ix} \text{ is 1-Lipschitz,}\\ &\le \frac{1}{(2\pi)^2}\int_{[-\pi,\pi]^2} |\omega-\xi|^2|\tau|^2|\widehat{\psi}(\omega)|^2 \mathrm{d}\omega && \text{by Cauchy-Schwarz,}\\ &= \sigma^2|\tau|^2,\end{aligned}$$

which leads to the desired result of eq. (6.3):

$$\|x_\tau * \psi - e^{-i\xi\cdot\tau}(x*\psi)\|_\infty \le \sigma\,|\tau|\,\|x\|_2.$$

E.2 Proof of equation (6.4)

We have

$$\mathrm{ReLU}(x*\psi_\alpha) = \mathrm{ReLU}(x*\mathrm{Re}(e^{-i\alpha}\psi)) = \mathrm{ReLU}(\mathrm{Re}(e^{-i\alpha}x*\psi)),$$

since x is real. By writing: $x * \psi = |x * \psi| e^{i\varphi(x*\psi)}$ where $\varphi(x*\psi)$ is the phase of $x*\psi$, this leads to

$$\begin{aligned}\mathrm{ReLU}(\mathrm{Re}(e^{-i\alpha}x * \psi)) &= \mathrm{ReLU}(\mathrm{Re}(|x*\psi|e^{i(\varphi(x*\psi)-\alpha)})) \\ &= \mathrm{ReLU}(|x*\psi|\cos(\varphi(x*\psi)-\alpha)) \\ &= |x*\psi|\,\mathrm{ReLU}(\cos(\varphi(x*\psi)-\alpha)),\end{aligned}$$

since ReLU activation is positive-homogeneous of degree 1. Thus,

$$\begin{aligned}\frac{1}{2}\int_{-\pi}^{\pi}\mathrm{ReLU}(x*\psi_\alpha)\mathrm{d}\alpha &= \frac{1}{2}\int_{-\pi}^{\pi}|x*\psi|\,\mathrm{ReLU}(\cos(\varphi(x*\psi)-\alpha))\mathrm{d}\alpha \\ &= \frac{1}{2}|x*\psi|\int_{-\pi-\varphi(x*\psi)}^{\pi-\varphi(x*\psi)}\mathrm{ReLU}(\cos(-\alpha))\mathrm{d}\alpha \qquad \text{with a change of variable,} \\ &= \frac{1}{2}|x*\psi|\int_{-\pi}^{\pi}\mathrm{ReLU}(\cos(\alpha))\mathrm{d}\alpha \qquad \text{since cos is } 2\pi \text{ periodic and even,} \\ &= \frac{1}{2}|x*\psi|\int_{-\pi/2}^{\pi/2}\cos(\alpha)\mathrm{d}\alpha \\ &= |x*\psi|.\end{aligned}$$

For $z \in \mathbb{C}$, we have $|z| = \sqrt{|\mathrm{Re}(z)|^2 + |\mathrm{Im}(z)|^2} \approx |\mathrm{Re}(z)| + |\mathrm{Im}(z)|$ in the following sense:

$$\frac{1}{\sqrt{2}}(|\mathrm{Re}(z)| + |\mathrm{Im}(z)|) \leq |z| \leq |\mathrm{Re}(z)| + |\mathrm{Im}(z)|.$$

We can write

$$\begin{aligned}|\mathrm{Re}(z)| &= \mathrm{ReLU}(\mathrm{Re}(z)) + \mathrm{ReLU}(-\mathrm{Re}(z)), \\ |\mathrm{Im}(z)| &= \mathrm{ReLU}(\mathrm{Im}(z)) + \mathrm{ReLU}(-\mathrm{Im}(z)).\end{aligned}$$

and then, using $\mathrm{Im}(z) = \mathrm{Re}(e^{i\pi/2}z)$ and $e^{i\pi} = -1$,

$$|z| \approx \mathrm{ReLU}(\mathrm{Re}(z)) + \mathrm{ReLU}(\mathrm{Re}(e^{-i\pi}z)) + \mathrm{ReLU}(\mathrm{Re}(e^{-i\pi/2}z)) + \mathrm{ReLU}(\mathrm{Re}(e^{i\pi/2}z)).$$

Finally,

$$|x*\psi| = \frac{1}{2}\int_{-\pi}^{\pi}\mathrm{ReLU}(x*\psi_\alpha)\mathrm{d}\alpha \approx \sum_{\alpha\in\{-\pi/2,0,\pi/2,\pi\}}\mathrm{ReLU}(\mathrm{Re}(x*\psi_\alpha)),$$

which shows that the integral can be well approximated with a sum of 4 phases α of the complex filter ψ.

E.3 Proof of Theorem 6.2

We first use the chain rule for the entropy:

$$H\Big(\varphi(D^{\mathrm{T}}x)\,\Big|\,\Big|D^{\mathrm{T}}x\Big|\Big) = H(\Big|D^{\mathrm{T}}x\Big|, \varphi(D^{\mathrm{T}}x)) - H(\Big|D^{\mathrm{T}}x\Big|).$$

The first term is rewritten with a change of variable:

$$\begin{aligned}H(\Big|D^{\mathrm{T}}x\Big|, \varphi(D^{\mathrm{T}}x)) &= H(D^{\mathrm{T}}x) - \sum_{k=1}^{d}\mathbb{E}[\log\Big|(D^{\mathrm{T}}x)_k\Big|] \\ &= H(x) - \sum_{k=1}^{d}\mathbb{E}[\log\Big|(D^{\mathrm{T}}x)_k\Big|] \qquad \text{as } D \text{ is unitary and hence } |\det(D)| = 1, \\ &\geq H(x) - d\mathbb{E}\left[\log\left(\frac{1}{d}\Big\|D^{\mathrm{T}}x\Big\|_1\right)\right] \qquad \text{by concavity,} \\ &\geq H(x) - d\log\left(\frac{1}{d}\mathbb{E}\left[\Big\|D^{\mathrm{T}}x\Big\|_1\right]\right) \qquad \text{by concavity.}\end{aligned}$$

The second term is bounded using the fact that the exponential distribution $\mathcal{E}(\lambda)$ is the maximum-entropy distribution on $\mathbb{R}_+$ with mean $\frac{1}{\lambda}$:

$$\begin{aligned} H(\left|D^{\mathrm{T}}x\right|) &\leq \sum_{k=1}^{d} H(\left|(D^{\mathrm{T}}x)_k\right|) \\ &\leq \sum_{k=1}^{d} \log\Big(e\mathbb{E}[\left|(D^{\mathrm{T}}x)_k\right|]\Big) \\ &\leq d\log\Big(\frac{e}{d}\mathbb{E}[\|D^{\mathrm{T}}x\|_1]\Big) \qquad \text{by concavity.} \end{aligned}$$

Combining both inequalities and rearranging terms yields the stated bound

$$H\Big(\varphi(D^{\mathrm{T}}x)\,\Big|\,\left|D^{\mathrm{T}}x\right|\Big) \geq H(x) - d - 2d\log\Big(\frac{1}{d}\mathbb{E}\Big[\left\|D^{\mathrm{T}}x\right\|_1\Big]\Big).$$

E.4 Proof of Theorem 6.3

We begin with the following lemma:

Lemma E.1. *Let $(\theta_1, \dots, \theta_d)$ be i.i.d. uniform random variables in $[0, 2\pi]$. Then there exists a constant $C_d > 0$ such that for all $(\rho_1, \dots, \rho_d) \in \mathbb{R}^d$, then*

$$\mathbb{E}\left[\left|\sum_{k=1}^{d} \rho_k e^{i\theta_k}\right|\right] \geq C_d \sqrt{\sum_{k=1}^{d} \rho_k^2}.$$

This is proved by observing that the left-hand side is a norm on $\mathbb{R}^d$. One can indeed verify that it is positive definite, homogeneous and satisfies the triangle inequality. Since all norms on $\mathbb{R}^d$ are equivalent, there exists a constant $C_d > 0$ such that

$$\mathbb{E}\left[\left|\sum_{k=1}^{d} \rho_k e^{i\theta_k}\right|\right] \geq C_d \sqrt{\sum_{k=1}^{d} \rho_k^2}.$$

for all $(\rho_1, \dots, \rho_d) \in \mathbb{R}^d$.

Going back to the proof of Theorem 6.3, and letting $x' = \rho_\lambda(D^{\mathrm{T}}x)$, we then have

$$\begin{aligned} \mathbb{E}\Big[\|D'^{\mathrm{T}}x'\|_1 \,\Big|\, |x'|\Big] &= \sum_{m=1}^{d} \mathbb{E}\left[\left|\sum_{k=1}^{d} D'_{k,m}x'_k\right| \,\middle|\, |x'|\right] \\ &\geq C_d \sum_{m=1}^{d} \sqrt{\sum_{k=1}^{d} |D'_{k,m}|^2 |x'_k|^2} \qquad \text{by the above lemma,} \\ &\geq C_d \sum_{m=1}^{d} \sum_{k=1}^{d} |D'_{k,m}|^2 |x'_k| \qquad \text{by concavity, because } \sum_{k=1}^{d} |D'_{k,m}|^2 = 1, \\ &= C_d \left\|x'\right\|_1 \qquad \text{because } \sum_{m=1}^{d} |D'_{k,m}|^2 = 1. \end{aligned}$$

Taking the expectation finishes the proof:

$$\mathbb{E}\Big[\|D'^{\mathrm{T}}x'\|_1\Big] \geq C_d \mathbb{E}\Big[\|x'\|_1\Big]. \tag{E.1}$$

	j	1	2	3	4	5	6	7	8	9	10	11
CIFAR-10	c_j	64	128	256	512	512	512	512	512	-	-	-
ImageNet	c_j	32	64	64	128	256	512	512	512	512	512	256

Table E.1: Number c_j of complex output channels of P_j, $1 \leq j \leq J$. The total number of projectors is $J = 8$ for CIFAR and $J = 11$ for ImageNet.

	PCScat	PCScat + skip	ResNet
CIFAR-10	41.6	83.1	0.27
ImageNet	36.0	62.8	11.7

Table E.2: Number of real parameters (in millions) of Learned Scattering network architectures. A complex parameter is counted as two real parameters.

E.5 Experimental details

Channel operators. In all experiments we set $P_0 = \mathrm{Id}$, and factorize the classifier with an additional complex 1×1 convolutional operator P_J, which reduces the dimension before all channels and positions are linearly combined. The architectures implemented are thus also written as $\prod_{j=1}^{J} P_j \rho W$, where ρ is the non-linearity. Each operator $(P_j)_{1 \leq j \leq J}$ is preceded by a standardization. It sets the complex mean $\mu = \mathbb{E}[z]$ of every channel to zero, and the real variance $\sigma^2 = \mathbb{E}[|z|^2]$ of every channel to one. This is similar to a complex 2D batch-normalization layer (Ioffe and Szegedy, 2015), but without learned affine parameters. Each operator $(P_j)_{1 \leq j \leq J}$ is additionally followed by a spatial divisive normalization (Wainwright et al., 2001a), similarly to the local response normalization of Krizhevsky et al. (2012). It sets the norm across channels of each spatial position to one. The sizes of the $(P_j)_j$ are specified in Table E.1.

The total numbers of parameters for each architecture are specified in Table E.2. Learned Scattering with phase collapse have a large number of parameters compared to ResNet, despite the comparable width. This is because the predefined wavelet operator W expands the dimension by a factor of $L + 1$, which means that the input dimension of the learned $(P_j)_j$ is higher than in ResNet. The skip-connection further increases this input dimension by a factor of 2.

Spatial filters. We use elongated Morlet filters for the L complex band-pass filters $(g_\ell)_\ell$ which are rotated versions of a mother wavelet g: $g_\ell(u) = g(r_{-\pi\ell/L}u)$, with r_θ the rotation by angle θ. The mother wavelet g is defined as

$$g(u) = \frac{\sigma^2}{2\pi/s^2}(e^{i\xi \cdot u} - K)e^{-u \cdot \Sigma u/2} \qquad \text{with } \Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 s^2 \end{pmatrix}, \tag{E.2}$$

Its parameters are its center frequency $\xi = ((3\pi/4)/2^\gamma, 0)$, its bandwidth $\sigma = 1.25 \times 2^{-\gamma}$, and its slant $s = 0.5$, where 2^γ designates the scale of the band-pass filter and is to be adjusted.

g is rotated along $L = 8$ angles for Imagenet and $L = 4$ angles for CIFAR: $\theta_\ell = (\pi\ell/L)_{1 \leq \ell \leq L}$. The $(g_\ell)_\ell$ are then discretized for numerical computations, and K is adjusted so that they have a zero mean.

Finally, we use for the low frequency g_0 a Gaussian window

$$g_0(u) = \frac{\sigma^2}{2\pi} e^{-\sigma^2 \|u\|_2^2/2}.$$

The filters are implemented with the *Kymatio* package (Andreux et al., 2020).

Intermediate scales $2^{j/2}$ are obtained by applying a subsampling by 2 after each block of 2 layers. This introduces intermediate scales and generates a wavelet filterbank with 2 scales per octave: the filters are designed so that when j low-pass filters and one band-pass filter are cascaded, with a subsampling every 2 layers, the scale of the resulting wavelet is $2^{j/2}$.

Each block comprises in its first layer a low-frequency filter g_0^1 with $\gamma = -1/2$ and band-pass filters with $\gamma = 0$. In the second layer, we use the same low-frequency filter $g_0^2 = g_0^1$ with $\gamma = -1/2$. The band-pass filters g_ℓ^2 are obtained with parameters $\xi' = (\pi/\sqrt{2}, 0)$, $\sigma' = 1.25\sqrt{2/3}$, and $s' = \sqrt{0.2}$.

For CIFAR experiments, the $J = 8$ layers are grouped in 4 successive blocks of 2 layers. For ImageNet experiments, the first layer consists of band-pass elongated Morlet filters g_ℓ and a low-pass Gaussian window g_0 with $\gamma = 0$, followed by a subsampling of 2. The 10 following layers are grouped in 5 blocks of 2 layers.

Optimization. We use the optimizer SGD with an initial learning rate of 0.01, a momentum of 0.9, a weight decay of 0.0001, and a batch size of 128. The classifier is preceded by a 2D batch-normalization layer. We use traditional data augmentation: horizontal flips and random crops for CIFAR, random resized crops of size 224 and horizontal flips for ImageNet. Classification error on ImageNet validation set is computed on a single center crop of size 224. On CIFAR, training lasts for 300 epochs and the learning rate is divided by 10 every 70 epochs. On ImageNet, training lasts for 150 epochs and the learning rate is divided by 10 every 45 epochs.

Appendix

F Appendix for Chapter 7

Chapter content

F.1 Proof of Theorem 7.1 165
F.1.1 Proof outline 165
F.1.2 Proof of Lemma F.1 168
F.1.3 Proof of Lemma F.2 169
F.1.4 Proof of Lemma F.3 169
F.1.5 Proof of Lemma F.4 171
F.2 Proof of Theorem 7.2 171
F.3 Proof of Theorem 7.3 173
F.4 Experimental details 174

F.1 Proof of Theorem 7.1

We prove a slightly more general version of Theorem 7.1 which we will need in the proof of Theorem 7.2. We allow the input x to be in a possibly infinite-dimensional Hilbert space H_0 (the finite-dimensional case is recovered with $H_0 = \mathbb{R}^{d_0}$). We shall assume that the random feature distribution π has bounded second- and fourth-order moments in the sense of Section 7.2.2: it admits a bounded uncentered covariance operator $C = \mathbb{E}_{w\sim\pi}[ww^{\mathrm{T}}]$ and $\mathbb{E}_{w\sim\pi}[(w^{\mathrm{T}}Tw)^2] < +\infty$ for every trace-class operator T on H_0. Without loss of generality, we assume that the non-linearity ρ is 1-Lipschitz and that $\rho(0) = 0$. These last assumptions simplify the constants involved in the analysis. They can be satisfied for any L-Lipschitz non-linearity ρ by replacing it with $(\rho - \rho(0))/L$, which does not change the linear expressivity of the network.

We give the proof outline in Appendix F.1.1. It relies on several lemmas, which are proven in Appendices F.1.2 to F.1.5. We write $\|\cdot\|_\infty$ the operator norm, $\|\cdot\|_2$ the Hilbert-Schmidt norm, and $\|\cdot\|_1$ the nuclear (or trace) norm.

F.1.1 Proof outline

The convergence of the activations $\hat{\varphi}(x)$ to the feature vector $\varphi(x)$ relies on the convergence of the empirical kernel $\hat{k}$ to the asymptotic kernel k. We thus begin by reformulating the mean-square error $\mathbb{E}_x[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2]$ in terms of the kernels $\hat{k}$ and k. More precisely, we will consider the integral operators $\hat{T}$ and T associated to the kernels. These integral operators are the infinite-dimensional equivalent of Gram matrices $(k(x_i, x_{i'}))_{1\le i,i'\le n}$.

Let μ be the distribution of x. We define the integral operator $T\colon L^2(\mu) \to L^2(\mu)$ associated to the asymptotic kernel k as

$$(Tf)(x) = \mathbb{E}_{x'}\Big[k(x, x')\, f(x')\Big],$$

where x' is an i.i.d. copy of x and μ is the law of x. Similarly, we denote $\hat{T}$ the integral operator defined by $\hat{k}$. Their standard properties are detailed in the next lemma. Moreover, the definition of $\hat{T}$ entails that it is the average of d_1 i.i.d. integral operators defined by the individual random features $(w_i)_{i\leq d_1}$ of $\hat{\varphi}$. The law of large numbers then implies a mean-square convergence of $\hat{T}$ to T, as proven in the following lemma.

Lemma F.1. *T and $\hat{T}$ are trace-class non-negative self-adjoint operators on $L^2(\mu)$, with*

$$\operatorname{tr}(T) \leq \|C\|_\infty \, \mathbb{E}_x[\|x\|^2].$$

The eigenvalues of T and $\hat{T}$ are the same as their respective activation covariance matrices $\mathbb{E}_x[\varphi(x)\,\varphi(x)^{\mathrm{T}}]$ and $\mathbb{E}_x[\hat{\varphi}(x)\,\hat{\varphi}(x)^{\mathrm{T}}]$. Besides, it holds that $\mathbb{E}_W[\hat{T}] = T$ and

$$\sqrt{\mathbb{E}_W\left[\|\hat{T} - T\|_2^2\right]} = \sqrt{\mathbb{E}_{W,x,x'}\left[|\hat{k}(x,x') - k(x,x')|^2\right]} = c\, d_1^{-1/2},$$

with some constant $c < +\infty$.

We defer the proof, which relies on standard properties and a direct calculation of the variance of $\hat{T}$ around its mean T, to Appendix F.1.2. In the following, we shall write $c = \kappa \, \|C\|_\infty \, \mathbb{E}_x[\|x\|^2]$ to simplify calculations for the proof of Theorem 7.2, where $C = \mathbb{E}_{w\sim\pi}[ww^{\mathrm{T}}]$ is the uncentered covariance of π, and κ is a constant. When π is Gaussian, Appendix F.1.2 further shows that $\kappa \leq \sqrt{3}$.

The mean-square error between $\hat{\varphi}$ and φ after alignment can then be expressed as a different distance between $\hat{T}$ and T, as proven in the next lemma.

Lemma F.2. *The alignment error between $\hat{\varphi}$ and φ is equal to the Bures-Wasserstein distance* BW *between $\hat{T}$ and T:*

$$\min_{\hat{A}\in\mathcal{O}(d_1)} \mathbb{E}_x\left[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2\right] = \mathrm{BW}(\hat{T}, T)^2.$$

The Bures-Wasserstein distance (Bhatia et al., 2019) is defined, for any trace-class non-negative self-adjoint operators $\hat{T}$ and T, as

$$\mathrm{BW}(\hat{T}, T)^2 = \min_{\hat{A}\in O\left(L^2(\mu)\right)} \|\hat{A}\,\hat{T}^{1/2} - T^{1/2}\|_2^2 = \operatorname{tr}\left(\hat{T} + T - 2\left(T^{1/2}\hat{T}T^{1/2}\right)^{1/2}\right).$$

The minimization in the first term is done over unitary operators of $L^2(\mu)$, and can be solved in closed-form with a singular value decomposition of $T^{1/2}\hat{T}^{1/2}$ as in eqs. (7.3) and (7.4). A direct calculation then shows that the minimal value is equal to the expression in the second term, as in eq. (7.5). The Bures-Wasserstein distance arises in optimal transport as the Wasserstein-2 distance between two zero-mean Gaussian distributions of respective covariance operators $\hat{T}$ and T, and in quantum information as the Bures distance, a non-commutative generalization of the Hellinger distance. We refer the interested reader to Bhatia et al. (2019) for more details. We defer the proof of Lemma F.2 to Appendix F.1.3.

It remains to establish the convergence of $\hat{T}$ towards T for the Bures-Wasserstein distance, which is a distance on the square roots of the operators. The main difficulty comes from the fact that the square root is Lipschitz only when bounded away from zero. This lack of regularity in the optimization problem can be seen from the fact that the optimal alignment rotation $\hat{A}$ is obtained by setting all singular values of some operator to one, which is unstable when this operator has vanishing singular values. We thus consider an entropic regularization of the underlying optimal transport problem over $\hat{A}$ with a parameter $\lambda > 0$ that will be adjusted with d_1. It penalizes the entropy of the coupling so that singular values smaller than λ are not amplified. It leads to a bound on the Bures-Wasserstein distance, as shown in the following lemma.

Lemma F.3. *Let $\hat{T}$ and T be two trace-class non-negative self-adjoint operators. For any $\lambda > 0$, we have*

$$\mathrm{BW}(\hat{T}, T)^2 \leq \frac{\|T\|_2 \|\hat{T} - T\|_2}{\lambda} + \operatorname{tr}(\hat{T} - T) + 2 \operatorname{tr}\Big(T + \lambda \operatorname{Id} - \big(T^2 + \lambda^2 \operatorname{Id}\big)^{1/2}\Big). \tag{F.1}$$

We defer the proof to Appendix F.1.4.

The first two terms in eq. (F.1) are controlled in expectation with Lemma F.1. The last term, when divided by λ, has a similar behavior to another quantity which arises in least-squares regression, namely the degrees of freedom $\operatorname{tr}(T(T + \lambda \operatorname{Id})^{-1})$ (Hastie and Tibshirani, 1987; Caponnetto and De Vito, 2007). It can be calculated by assuming a decay rate for the eigenvalues of T, as done in the next lemma.

Lemma F.4. *Let T be a trace-class non-negative self-adjoint operator whose eigenvalues satisfy $\lambda_m \leq c\, m^{-\alpha}$ for some $\alpha > 1$ and $c > 0$. Then it holds:*

$$\operatorname{tr}\Big(T + \lambda \operatorname{Id} - \big(T^2 + \lambda^2 \operatorname{Id}\big)^{1/2}\Big) \leq c'\, \lambda^{1 - 1/\alpha},$$

where the constant $c' = \frac{c^{1/\alpha}}{1 - 1/\alpha}$.

The proof is in Appendix F.1.5.

We now put together Lemmas F.1 to F.4. We have for any $\lambda > 0$,

$$\mathbb{E}_{W,x}\Big[\|\hat{A}\, \hat{\varphi}(x) - \varphi(x)\|_H^2\Big] = \mathbb{E}_W\Big[\mathrm{BW}(\hat{T}, T)^2\Big] \leq \frac{\kappa\, \|C\|_\infty^2 \mathbb{E}_x[\|x\|^2]^2}{\lambda \sqrt{d_1}} + \frac{2c^{1/\alpha}}{1 - 1/\alpha} \lambda^{1 - \frac{1}{\alpha}},$$

where we have used the Cauchy-Schwarz inequality to bound $\mathbb{E}_W[\|\hat{T} - T\|_2] \leq \sqrt{\mathbb{E}_W[\|\hat{T} - T\|_2^2]}$ and the fact that $\|T\|_2 \leq \operatorname{tr} T \leq \|C\|_\infty \mathbb{E}_x[\|x\|^2]$. We then optimize the upper bound with respect to λ by setting

$$\lambda = \left(\frac{2c^{1/\alpha}\sqrt{d_1}}{\kappa \|C\|_\infty^2 \mathbb{E}_x[\|x\|^2]^2}\right)^{-\alpha/(2\alpha - 1)},$$

which yields

$$\mathbb{E}_{W,x}\Big[\|\hat{A}\, \hat{\varphi}(x) - \varphi(x)\|_H^2\Big] \leq c''\, d_1^{-(\alpha-1)/(4\alpha-2)},$$

with a constant

$$c'' = \frac{2\kappa^{(\alpha-1)/(2\alpha-1)}}{(\alpha - 1)/(2\alpha - 1)} \left(\frac{c}{\|C\|_\infty \mathbb{E}_x[\|x\|^2]}\right)^{1/(2\alpha-1)} \|C\|_\infty\, \mathbb{E}_x[\|x\|^2].$$

Finally, the function $\hat{f}$ can be written

$$\hat{f}(x) = \langle \hat{A}^{\mathrm{T}} \theta, \hat{\varphi}(x) \rangle = \langle \theta, \hat{A}\, \hat{\varphi}(x) \rangle_H,$$

so that

$$|\hat{f}(x) - f(x)|^2 = |\langle \theta, \hat{A}\, \hat{\varphi}(x) - \varphi(x) \rangle_H|^2 \leq \|\theta\|_H^2 \|\hat{A}\, \hat{\varphi}(x) - \varphi(x)\|_H^2.$$

Rewriting $\|\theta\|_H = \|f\|_{\mathcal{H}}$, assuming that θ is the minimum-norm vector such that $f(x) = \langle \theta, \varphi(x) \rangle_H$, and using the convergence of $\hat{A}\, \hat{\varphi}$ towards φ then yields

$$\mathbb{E}_{W,x}\Big[|\hat{f}(x) - f(x)|^2\Big] \leq c'' \|f\|_{\mathcal{H}}^2\, d_1^{-(\alpha-1)/(4\alpha-2)}.$$

F.1.2 Proof of Lemma F.1

We define the linear operator $\Phi\colon L^2(\mu) \to H$ by

$$\Phi f = \mathbb{E}_x[f(x)\,\varphi(x)].$$

Its adjoint $\Phi^{\mathrm{T}}\colon H \to L^2(\mu)$ is then given by

$$(\Phi^{\mathrm{T}} u)(x) = \langle u, \varphi(x)\rangle,$$

so that $T = \Phi^{\mathrm{T}}\Phi$. This proves that T is self-adjoint and non-negative. On the other hand, we have $\Phi\Phi^{\mathrm{T}} = \mathbb{E}_x[\varphi(x)\,\varphi(x)^{\mathrm{T}}]$ the uncentered covariance matrix of the feature map φ associated to the kernel k. This shows that T and this uncentered covariance matrix have the same eigenvalues.

Moreover, we have

$$\operatorname{tr}(T) = \mathbb{E}_x[k(x,x)] = \operatorname{tr}\Big(\Phi^{\mathrm{T}}\Phi\Big) = \|\Phi\|_2^2 = \mathbb{E}_x\Big[\|\varphi(x)\|^2\Big],$$

and using the definition of k,

$$\mathbb{E}_x[k(x,x)] = \mathbb{E}_{x,w}\Big[\rho(\langle x,w\rangle)^2\Big] \le \mathbb{E}_{x,w}\Big[|\langle x,w\rangle|^2\Big] = \operatorname{tr}\Big(C\,\mathbb{E}_x\Big[xx^{\mathrm{T}}\Big]\Big) \le \|C\|_\infty\,\mathbb{E}_x\Big[\|x\|^2\Big],$$

where $w \sim \pi$ independently from x, $|\rho(t)| \le |t|$ by assumption on ρ, and the last step follows from Hölder's inequality. This proves that T is trace-class and Φ is Hilbert-Schmidt, with an explicit upper bound on the trace.

The above remarks are also valid for $\hat{T}$ with an appropriate definition of $\hat{\Phi}\colon L^2(\mu) \to \mathbb{R}^{d_1}$. We have $\mathbb{E}_W[\hat{T}] = T$ because $\mathbb{E}_W[\hat{k}(x,x')] = k(x,x')$. Therefore, $\operatorname{tr}(\hat{T}) = \|\hat{\Phi}\|_2^2$ is almost surely finite because

$$\mathbb{E}_W[\operatorname{tr}(\hat{T})] = \operatorname{tr}(T) < +\infty.$$

Let $\hat{k}_i(x,x') = \rho(\langle x, w_i\rangle)\,\rho(\langle x', w_i\rangle)$ where $(w_i)_{i\le d_j}$ are the rows of W, and $\hat{T}_i$ the associated integral operators. The $\hat{T}_i$ are i.i.d. with $\mathbb{E}_W[\hat{T}_i] = T$ as for $\hat{T}$, and we have $\hat{T} = d_1^{-1}\sum_{i=1}^{d_1}\hat{T}_i$. It then follows by standard variance calculations that

$$\mathbb{E}_W\Big[\|\hat{T} - T\|_2^2\Big] = \frac{1}{d_1}\Big(\mathbb{E}_W\Big[\|\hat{T}_1\|_2^2\Big] - \|T\|_2^2\Big) = \frac{c}{d_1},$$

with a constant c such that

$$c \le \mathbb{E}_W\Big[\|\hat{T}_1\|_2^2\Big] \le \mathbb{E}_W\Big[\operatorname{tr}(\hat{T}_1)^2\Big] = \mathbb{E}_W\Big[\mathbb{E}_x\Big[\rho(\langle x,w_1\rangle)^2\Big]^2\Big] \le \mathbb{E}_W\Big[\mathbb{E}_x\Big[|\langle x,w_1\rangle|^2\Big]^2\Big].$$

We then have, using the assumption on the fourth moments of π,

$$\mathbb{E}_W\Big[\mathbb{E}_x\Big[|\langle x,w_1\rangle|^2\Big]^2\Big] = \mathbb{E}_W\Big[\Big(w_1^{\mathrm{T}}\mathbb{E}_x\Big[xx^{\mathrm{T}}\Big]w_1\Big)^2\Big] < +\infty,$$

because $\operatorname{tr}\mathbb{E}_x[xx^{\mathrm{T}}] = \mathbb{E}_x[\|x\|^2] < +\infty$. When π is Gaussian, we further have

$$\begin{aligned}
\mathbb{E}_W\Big[\mathbb{E}_x\Big[|\langle x,w_1\rangle|^2\Big]^2\Big] &= \Big(\operatorname{tr}\Big(C\,\mathbb{E}_x\Big[xx^{\mathrm{T}}\Big]\Big)\Big)^2 + 2\operatorname{tr}\Big(\Big(C\,\mathbb{E}_x\Big[xx^{\mathrm{T}}\Big]\Big)^2\Big)\\
&\le 3\Big(\operatorname{tr}\Big(C\,\mathbb{E}_x\Big[xx^{\mathrm{T}}\Big]\Big)\Big)^2\\
&\le 3\,\|C\|_\infty^2\,\mathbb{E}_x\Big[\|x\|^2\Big]^2,
\end{aligned}$$

by classical fourth-moment computations of Gaussian random variables.

F.1.3 Proof of Lemma F.2

The alignment error can be rewritten in terms of the linear operators Φ and $\hat{\Phi}$ defined in Appendix F.1.2:

$$\mathbb{E}_x\Big[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2\Big] = \|\hat{A}\,\hat{\Phi} - \Phi\|_2^2.$$

We then expand

$$\|\hat{A}\,\hat{\Phi} - \Phi\|_2^2 = \|\hat{\Phi}\|_2^2 + \|\Phi\|_2^2 - 2\operatorname{tr}\Big(\Phi^{\mathrm{T}}\hat{A}\hat{\Phi}\Big).$$

The first two terms are respectively equal to $\operatorname{tr}\hat{T}$ and $\operatorname{tr}T$ per Appendix F.1.2. The alignment error is minimized with $\hat{A} = UV^{\mathrm{T}}$ from the SVD decomposition (Bhatia et al., 2019):

$$\Phi\hat{\Phi}^{\mathrm{T}} = \mathbb{E}_x\Big[\varphi(x)\,\hat{\varphi}(x)^{\mathrm{T}}\Big] = USV^{\mathrm{T}},$$

for which we then have

$$\operatorname{tr}\Big(\Phi^{\mathrm{T}}\hat{A}\hat{\Phi}\Big) = \operatorname{tr}\Big(\hat{\Phi}\Phi^{\mathrm{T}}\hat{A}\Big) = \operatorname{tr}\Big(VSU^{\mathrm{T}}UV^{\mathrm{T}}\Big) = \operatorname{tr}(S).$$

This can further be written

$$\operatorname{tr}(S) = \operatorname{tr}\Big(\big(US^2U^{\mathrm{T}}\big)^{1/2}\Big) = \operatorname{tr}\Big(\big(\Phi\hat{\Phi}^{\mathrm{T}}\hat{\Phi}\Phi^{\mathrm{T}}\big)^{1/2}\Big) = \operatorname{tr}\Big(\big(\Phi\hat{T}\Phi^{\mathrm{T}}\big)^{1/2}\Big).$$

To rewrite this in terms of T, we perform a polar decomposition of Φ: there exists a unitary operator $P\colon L^2(\mu) \to H$ such that $\Phi = PT^{1/2}$. We then have

$$\begin{aligned}
\operatorname{tr}\Big(\big(\Phi\hat{T}\Phi^{\mathrm{T}}\big)^{1/2}\Big) &= \operatorname{tr}\Big(\big(PT^{1/2}\hat{T}T^{1/2}P^{\mathrm{T}}\big)^{1/2}\Big) \\
&= \operatorname{tr}\Big(P\big(T^{1/2}\hat{T}T^{1/2}\big)^{1/2}P^{\mathrm{T}}\Big) \\
&= \operatorname{tr}\Big(\big(T^{1/2}\hat{T}T^{1/2}\big)^{1/2}\Big).
\end{aligned}$$

Putting everything together, we have

$$\mathbb{E}_x\Big[\|\hat{A}\,\hat{\varphi}(x) - \varphi(x)\|_H^2\Big] = \operatorname{tr}\Big(\hat{T} + T - 2\big(T^{1/2}\hat{T}T^{1/2}\big)^{1/2}\Big).$$

F.1.4 Proof of Lemma F.3

The Bures-Wasserstein distance can be rewritten as a minimum over contractions rather than unitary operators:

$$\mathrm{BW}(\hat{T}, T)^2 = \min_{\|\hat{A}\|_\infty \le 1} \operatorname{tr}\Big(\hat{T} + T - 2T^{1/2}\hat{A}\hat{T}^{1/2}\Big),$$

which holds because of Hölder's inequality:

$$\operatorname{tr}\Big(T^{1/2}\hat{A}\hat{T}^{1/2}\Big) = \operatorname{tr}\Big(\hat{T}^{1/2}T^{1/2}\hat{A}\Big) \le \|\hat{T}^{1/2}T^{1/2}\|_1\|\hat{A}\|_\infty = \operatorname{tr}\Big(\big(T^{1/2}\hat{T}T^{1/2}\big)^{1/2}\Big)\|\hat{A}\|_\infty.$$

Rather than optimizing over contractions $\hat{A}$, which leads to a unitary $\hat{A}$, we shall use a non-unitary $\hat{A}$ with $\|\hat{A}\|_\infty < 1$.

We introduce an "entropic" regularization: let $\lambda > 0$, and define

$$\mathrm{BW}_\lambda(\hat{T}, T)^2 = \min_{\|\hat{A}\|_\infty \leq 1} \mathrm{tr}\Big(\hat{T} + T - 2T^{1/2}\hat{A}\hat{T}^{1/2}\Big) + \lambda \log\det\bigg(\Big(\mathrm{Id} - \hat{A}^{\mathrm{T}}\hat{A}\Big)^{-1}\bigg).$$

The second term corresponds to the negentropy of the coupling in the underlying optimal transport formulation of the Bures-Wasserstein distance. It can be minimized in closed-form by calculating the fixed-point of Sinkhorn iterations (Janati et al., 2020), or with a direct SVD calculation as in Appendix F.1.3. It is indeed clear that the minimum is attained at some $\hat{A}_\lambda = US_\lambda V^{\mathrm{T}}$ with $T^{1/2}\hat{T}^{1/2} = USV^{\mathrm{T}}$, and this becomes a separable quadratic problem over the singular values S_λ. We thus find

$$\begin{aligned} S_\lambda &= \Big(\big(S^2 + \lambda^2 \mathrm{Id}\big)^{1/2} - \lambda \mathrm{Id}\Big) S^{-1}, \\ \hat{A}_\lambda &= \Big(\big(T^{1/2}\hat{T}T^{1/2} + \lambda^2 \mathrm{Id}\big)^{1/2} - \lambda \mathrm{Id}\Big) T^{-\frac{1}{2}}\hat{T}^{-\frac{1}{2}}, \end{aligned}$$

and one can verify that we indeed have $\|\hat{A}_\lambda\|_\infty < 1$. When plugged in the original distance, it gives the following upper bound:

$$\mathrm{BW}(\hat{T}, T)^2 \leq \mathrm{tr}\bigg(\hat{T} + T - 2\Big(\big(T^{1/2}\hat{T}T^{1/2} + \lambda^2 \mathrm{Id}\big)^{1/2} - \lambda \mathrm{Id}\Big)\bigg).$$

The term $\lambda^2 \mathrm{Id}$ in the square root makes this a Lipschitz function of $\hat{T}$. Indeed, define the function g by

$$g(\hat{T}) = \mathrm{tr}\Big(\big(T^{1/2}\hat{T}T^{1/2} + \lambda^2 \mathrm{Id}\big)^{1/2} - \lambda \mathrm{Id}\Big).$$

Standard calculations (Bhatia et al., 2019; Janati et al., 2020) then show that

$$\nabla g(\hat{T}) = \frac{1}{2} T^{1/2}\big(T^{1/2}\hat{T}T^{1/2} + \lambda^2 \mathrm{Id}\big)^{-1/2} T^{1/2}.$$

It implies that

$$0 \preccurlyeq \nabla g(\hat{T}) \preccurlyeq \frac{1}{2\lambda} T,$$

where have used that $T^{1/2}\hat{T}T^{1/2} \succcurlyeq 0$ in the second inequality, and finally,

$$\|\nabla g(\hat{T})\|_2 \leq \frac{\|T\|_2}{2\lambda}.$$

This last inequality follows from

$$\|\nabla g(\hat{T})\|_2^2 = \mathrm{tr}\Big(\nabla g(\hat{T})^{\mathrm{T}} \nabla g(\hat{T})\Big) \leq \mathrm{tr}\Big(\nabla g(\hat{T})^{\mathrm{T}} \frac{1}{2\lambda} T\Big) \leq \|\nabla g(\hat{T})\|_2 \frac{\|T\|_2}{2\lambda},$$

where we have used the operator-monotonicity of the map $M \mapsto \mathrm{tr}\Big(\nabla g(\hat{T})^{\mathrm{T}} M\Big)$, which holds because $\nabla g(\hat{T}) \succcurlyeq 0$.

Using the bound on the Lipschitz constant of g, we can then write

$$|g(\hat{T}) - g(T)| \leq \frac{\|T\|_2}{2\lambda} \|\hat{T} - T\|_2.$$

This leads to an inequality on the Bures-Wasserstein distance:

$$\begin{aligned} \mathrm{BW}(\hat{T}, T)^2 &\leq \mathrm{tr}(\hat{T} + T) - 2g(\hat{T}) \\ &= 2(\mathrm{tr}(T) - g(T)) + \mathrm{tr}(\hat{T} - T) - 2(g(\hat{T}) - g(T)) \\ &\leq 2(\mathrm{tr}(T) - g(T)) + \mathrm{tr}(\hat{T} - T) + \frac{\|T\|_2}{\lambda} \|\hat{T} - T\|_2, \end{aligned}$$

which concludes the proof.

F.1.5 Proof of Lemma F.4

We have

$$\operatorname{tr}\left(T + \lambda \operatorname{Id} - \left(T^2 + \lambda^2 \operatorname{Id}\right)^{1/2}\right) = \sum_{m=1}^{\infty} \left(\lambda_m + \lambda - \sqrt{\lambda_m^2 + \lambda^2}\right)$$

We have the following inequality

$$\lambda_m + \lambda - \sqrt{\lambda_m^2 + \lambda^2} \leq \min(\lambda_m, \lambda),$$

by using $\sqrt{\lambda_m^2 + \lambda^2} \geq \max(\lambda_m, \lambda)$.

We have $\lambda_m \leq c\, m^{-\alpha}$ for all m. We split the sum at $M = \lfloor (\lambda/c)^{-1/\alpha} \rfloor$ (so that $c\, M^{-\alpha} \approx \lambda$), and we have

$$\begin{aligned}
\sum_{m=1}^{M} \left(\lambda_m + \lambda - \sqrt{\lambda_m^2 + \lambda^2}\right) &\leq \sum_{m=1}^{M} \lambda = M\lambda, \\
\sum_{m=M+1}^{\infty} \left(\lambda_m + \lambda - \sqrt{\lambda_m^2 + \lambda^2}\right) &\leq \sum_{m=M+1}^{\infty} \lambda_m \leq c \sum_{m=M+1}^{\infty} m^{-\alpha} \leq c \frac{M^{1-\alpha}}{\alpha - 1},
\end{aligned}$$

Finally,

$$\sum_{1=1}^{\infty} \left(\lambda_m + \lambda - \sqrt{\lambda_m^2 + \lambda^2}\right) \leq \left(\frac{\lambda}{c}\right)^{-1/\alpha} \lambda + \frac{c}{\alpha - 1} \left(\frac{\lambda}{c}\right)^{1-1/\alpha} = \frac{c^{1/\alpha}}{1 - 1/\alpha}\, \lambda^{1-1/\alpha}.$$

F.2 Proof of Theorem 7.2

In this section, expectations are taken with respect to both the weights $W_1, \ldots, W_j$ and the input x. We remind that $W_j = W_j' \hat{A}_{j-1}$ with W_j' having i.i.d. rows $w_{ji}' \sim \pi_j$. Let $C_j = \mathbb{E}_{w_j \sim \pi_j}[w_j w_j^{\mathrm{T}}]$ be the uncentered covariance of π_j. Similarly to Appendix F.1, we assume without loss of generality that ρ is 1-Lipschitz and that $\rho(0) = 0$.

Let $\tilde{\phi}_j = \rho W_j' \phi_{j-1}$. Let $A_j \in \mathcal{O}(d_j)$ to be adjusted later. We have by definition of $\hat{A}_j$:

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\|\hat{A}_j\, \hat{\phi}_j(x) - \phi_j(x)\|^2\right]} &\leq \sqrt{\mathbb{E}\left[\|A_j\, \hat{\phi}_j(x) - \phi_j(x)\|^2\right]} \\
&\leq \sqrt{\mathbb{E}\left[\|A_j \hat{\phi}_j(x) - A_j \tilde{\phi}_j(x)\|^2\right]} + \sqrt{\mathbb{E}\left[\|A_j \tilde{\phi}_j(x) - \phi_j(x)\|^2\right]}, \quad \text{(F.2)}
\end{aligned}$$

where the last step follows by the triangle inequality. We now bound separately each term.

To bound the first term, we compute the Lipschitz constant of $\rho W_j'$ (in expectation). For any z, $z' \in H_{j-1}$, we have:

$$\begin{aligned}
\mathbb{E}\left[\|\rho W_j' z - \rho W_j' z'\|^2\right] &\leq \frac{1}{d_j} \mathbb{E}\left[\|W_j'(z - z')\|^2\right] \\
&= \frac{1}{d_j} \sum_{i=1}^{d_j} \mathbb{E}\left[|\langle z - z', w_{ji}' \rangle|^2\right] \\
&= (z - z')^{\mathrm{T}} C_j (z - z') \\
&\leq \|C_j\|_\infty \|z - z'\|^2,
\end{aligned}$$

where we have used the fact that ρ is 1-Lipschitz, and have made explicit the normalization factor of d_j^{-1}. We can therefore bound the first term in eq. (F.2):

$$\begin{aligned}\sqrt{\mathbb{E}\Big[\|A_j\hat{\phi}_j(x) - A_j\tilde{\phi}_j(x)\|^2\Big]} &= \sqrt{\mathbb{E}\Big[\|(\rho W_j')\hat{A}_{j-1}\hat{\phi}_{j-1}(x) - (\rho W_j')\phi_{j-1}(x)\|^2\Big]} \\ &\leq \|C_j\|_\infty^{1/2}\sqrt{\mathbb{E}\Big[\|\hat{A}_{j-1}\hat{\phi}_{j-1}(x) - \phi_{j-1}(x)\|^2\Big]}.\end{aligned}$$

We define A_j, which was arbitrary, as the minimizer of the second term in eq. (F.2) over $\mathcal{O}(d_j)$. We can then apply Theorem 7.1 to $z = \phi_{j-1}(x)$. Indeed, $\mathbb{E}_z[\varphi_j(z)\,\varphi_j(z)^{\mathrm{T}}] = \mathbb{E}_x[\phi_j(x)\,\phi_j(x)^{\mathrm{T}}]$ is trace-class with eigenvalues $\lambda_{j,m} = O(m^{-\alpha_j})$, and π_j has bounded second- and fourth-order moments. Therefore, there exists a constant c_j such that

$$\begin{aligned}\sqrt{\mathbb{E}\Big[\|A_j\tilde{\phi}_j(x) - \phi_j(x)\|^2\Big]} &= \sqrt{\mathbb{E}\Big[\|A_j\,\rho W_j'\,\phi_{j-1}(x) - \varphi_j\,\phi_{j-1}(x)\|^2\Big]} \\ &\leq \|C_j\|_\infty^{1/2}\sqrt{\mathbb{E}[\|\phi_{j-1}(x)\|^2]}\,c_j\,d_j^{-\eta_j/2},\end{aligned}$$

with $\eta_j = \frac{\alpha_j - 1}{2(2\alpha_j - 1)}$. We have made explicit the factors $\|C_j\|_\infty^{1/2}\sqrt{\mathbb{E}[\|\phi_{j-1}(x)\|^2]}$ in the constant coming from Theorem 7.1to simplify the expressions in the sequel. We can further bound $\sqrt{\mathbb{E}[\|\phi_{j-1}(x)\|^2]}$ by iteratively applying Lemma F.1 from Appendix F.1:

$$\sqrt{\mathbb{E}[\|\phi_{j-1}(x)\|^2]} \leq \|C_{j-1}\|_\infty^{1/2}\cdots\|C_1\|_\infty^{1/2}\sqrt{\mathbb{E}[\|x\|^2]}.$$

We thus have shown:

$$\begin{aligned}\sqrt{\mathbb{E}\Big[\|\hat{A}_j\,\hat{\phi}_j(x) - \phi_j(x)\|^2\Big]} \leq \|C_j\|_\infty^{1/2}&\sqrt{\mathbb{E}\Big[\|\hat{A}_{j-1}\hat{\phi}_{j-1}(x) - \phi_{j-1}(x)\|^2\Big]} \\ &+ \|C_j\|_\infty^{1/2}\cdots\|C_1\|_\infty^{1/2}\sqrt{\mathbb{E}[\|x\|^2]}\,c_j\,d_j^{-\eta_j/2}.\end{aligned}$$

It then follows by induction:

$$\sqrt{\mathbb{E}\Big[\|\hat{A}_j\,\hat{\phi}_j(x) - \phi_j(x)\|^2\Big]} \leq \|C_j\|_\infty^{1/2}\cdots\|C_1\|_\infty^{1/2}\sqrt{\mathbb{E}[\|x\|^2]}\sum_{\ell=1}^{j} c_\ell\,d_\ell^{-\eta_\ell/2}.$$

We conclude like in the proof of Theorem 7.1:

$$\sqrt{\mathbb{E}\Big[|\hat{f}(x) - f(x)|^2\Big]} \leq \|f\|_{\mathcal{H}_J}\|C_J\|_\infty^{1/2}\cdots\|C_1\|_\infty^{1/2}\sqrt{\mathbb{E}[\|x\|^2]}\sum_{j=1}^{J} c_j\,d_j^{-\eta_j/2}.$$

We finally show the convergence of the kernels. Let $\tilde{k}_j$ be the kernel defined by the feature map $\tilde{\phi}_j$. Expectations are now also taken with respect to x', an i.i.d. copy of x. We have by the triangle inequality:

$$|\hat{k}_j(x,x') - k_j(x,x')| \leq |\hat{k}_j(x,x') - \tilde{k}_j(x,x')| + |\tilde{k}_j(x,x') - k_j(x,x')|. \tag{F.3}$$

For the first term on the right-hand side:

$$\begin{aligned}|\hat{k}_j(x,x') - \tilde{k}_j(x,x')| &= |\langle\hat{\phi}_j(x),\hat{\phi}_j(x')\rangle - \langle\tilde{\phi}_j(x),\tilde{\phi}_j(x')\rangle| \\ &\leq |\langle\hat{\phi}_j(x),\hat{\phi}_j(x') - \tilde{\phi}_j(x')\rangle + \langle\hat{\phi}_j(x) - \tilde{\phi}_j(x),\tilde{\phi}_j(x')\rangle| \\ &\leq \|\hat{\phi}_j(x)\|\|\hat{\phi}_j(x') - \tilde{\phi}_j(x')\| + \|\tilde{\phi}_j(x')\|\|\hat{\phi}_j(x) - \tilde{\phi}_j(x)\|.\end{aligned}$$

We thus have, because x, x' are i.i.d.,

$$\sqrt{\mathbb{E}\Big[|\hat{k}_j(x,x') - \tilde{k}_j(x,x')|^2\Big]}$$
$$\leq \sqrt{\mathbb{E}\Big[\|\hat{\phi}_j(x)\|^2\Big]\mathbb{E}\Big[\|\hat{\phi}_j(x') - \tilde{\phi}_j(x')\|^2\Big]} + \sqrt{\mathbb{E}\Big[\|\tilde{\phi}_j(x')\|^2\Big]\mathbb{E}\Big[\|\hat{\phi}_j(x) - \tilde{\phi}_j(x)\|^2\Big]}.$$

Using the Lipschitz constant of $\rho W_j'$ in expectation as above:

$$\sqrt{\mathbb{E}\Big[|\hat{k}_j(x,x') - \tilde{k}_j(x,x')|^2\Big]} \leq 2\|C_j\|_\infty \sqrt{\mathbb{E}\Big[\|\phi_{j-1}(x)\|^2\Big]\mathbb{E}\Big[\|\hat{\phi}_{j-1}(x) - \phi_{j-1}(x)\|^2\Big]}.$$

The factors on the right-hand side can be bounded using the above, to yield

$$\sqrt{\mathbb{E}\Big[|\hat{k}_j(x,x') - \tilde{k}_j(x,x')|^2\Big]} \leq 2\|C_j\|_\infty \cdots \|C_1\|_\infty \mathbb{E}[\|x\|^2] \sum_{\ell=1}^{j-1} c_\ell \, d_\ell^{-\eta_\ell/2}.$$

The second term on the right-hand side of eq. (F.3) can be bounded with Theorem 7.1 applied to $z = \phi_{j-1}(x)$ as before:

$$\sqrt{\mathbb{E}\Big[|\tilde{k}_j(x,x') - k_j(x,x')|^2\Big]} \leq \kappa_j \|C_j\|_\infty \cdots \|C_1\|_\infty \mathbb{E}[\|x\|^2] \, d_j^{-1/2}.$$

where we have again used the upper bound on $\sqrt{\mathbb{E}[\|\phi_{j-1}(x)\|^2]}$.

We thus have shown that

$$\sqrt{\mathbb{E}\Big[|k_j(x,x') - k_j(x,x')|^2\Big]} \leq \|C_j\|_\infty \cdots \|C_1\|_\infty \mathbb{E}[\|x\|^2] \left(2\sum_{\ell=1}^{j-1} c_\ell \, d_\ell^{-\eta_\ell/2} + \kappa_j \, d_j^{-1/2}\right).$$

F.3 Proof of Theorem 7.3

We prove the result by induction on the layer index j. We initialize with $\phi_0(x) = x$, which admits an orthogonal representation $\sigma_0(g) = g$. Now suppose that ϕ_{j-1} admits an orthogonal representation σ_{j-1}. Let $w \sim \pi_j$, we have that $\sigma_{j-1}(g)^{\mathrm{T}} w \sim \pi_j$ for all $g \in G$ by hypothesis. When $\pi_j = \mathcal{N}(0, C_j)$, this is equivalent to $\sigma_{j-1}(g)^{\mathrm{T}} C_j \sigma_{j-1}(g) = C_j$, i.e. $\sigma_{j-1}(g) C_j = C_j \sigma_{j-1}(g)$. We begin by showing that ϕ_j then admits an orthogonal representation σ_j.

We have

$$\phi_j(gx) = \varphi_j(\phi_{j-1}(gx)) = \varphi_j(\sigma_{j-1}(g)\phi_{j-1}(x)).$$

For simplicity, here we define the feature map φ_j with $\varphi_j(z)(w) = \rho(\langle z, w\rangle)$ with $H_j = L^2(\pi_j)$ (the result of the theorem does however not depend on this choice, as all feature maps are related by a rotation). Then,

$$\phi_j(gx)(w) = \rho(\langle \sigma_{j-1}(g)\phi_{j-1}(x), w\rangle) = \rho(\langle \phi_{j-1}(x), \sigma_{j-1}(g)^{\mathrm{T}} w\rangle).$$

For each $g \in G$, we thus define the operator $\sigma_j(g)$ by its action on $\psi \in H_j$:

$$(\sigma_j(g)\psi)(w) = \psi(\sigma_{j-1}(g)^{\mathrm{T}} w).$$

It is obviously linear, and bounded as $\|\sigma_j(g)\|_\infty = 1$:

$$\|\sigma_j(g)\psi\|_{H_j}^2 = \mathbb{E}_w\Big[\psi(\sigma_{j-1}(g)^{\mathrm{T}} w)^2\Big] = \mathbb{E}_w\Big[\psi(w)^2\Big] = \|\psi\|_{H_j}^2,$$

where we have used that $\sigma_{j-1}(g)^{\mathrm{T}} w \sim w$. We further verify that $\sigma_j(gg') = \sigma_j(g)\sigma_j(g')$:

$$\begin{aligned}(\sigma_j(gg')\psi)(w) &= \psi(\sigma_{j-1}(gg')^{\mathrm{T}} w) = \psi(\sigma_{j-1}(g')^{\mathrm{T}}\sigma_{j-1}(g)^{\mathrm{T}} w) \\ &= (\sigma_j(g')\psi)(\sigma_{j-1}(g)^{\mathrm{T}} w) = (\sigma_j(g)\sigma_j(g')\psi)(w).\end{aligned}$$

We can thus write $\phi_j(gx) = \sigma_j(g)\phi_j(x)$, which shows that ϕ_j admits a representation.

It remains to show that $\sigma_j(g)$ is orthogonal. The adjoint $\sigma_j(g)^{\mathrm{T}}$ is equal to $\sigma_j(g^{\mathrm{T}})$:

$$\left\langle \sigma_j(g)\psi, \psi' \right\rangle_{H_j} = \mathbb{E}_w\left[\psi(\sigma_{j-1}(g)^{\mathrm{T}} w)\psi'(w)\right] = \mathbb{E}_w\left[\psi(w)\psi'(\sigma_{j-1}(g)w)\right] = \left\langle \psi, \sigma_j(g^{\mathrm{T}})\psi' \right\rangle_{H_j},$$

where we have used $\sigma_{j-1}(g)^{\mathrm{T}} = \sigma_{j-1}(g^{\mathrm{T}})$ since σ_{j-1} is a group homomorphism. It is then straightforward that $\sigma_j(g)\sigma_j(g)^{\mathrm{T}} = \sigma_j(g)^{\mathrm{T}}\sigma_j(g) = \mathrm{Id}$ by using again the fact that σ_j is a group homomorphism. This proves that $\sigma_j(g) \in O(H_j)$.

We finally show that the rainbow kernel k_j is invariant. We have

$$\begin{aligned}k_j(gx, gx') &= \left\langle \phi_j(gx), \phi_j(gx') \right\rangle_{H_j} = \left\langle \sigma_j(g)\phi_j(x), \sigma_j(g)\phi_j(x') \right\rangle_{H_j} \\ &= \left\langle \phi_j(x), \phi_j(x') \right\rangle_{H_j} = k_j(x, x'),\end{aligned}$$

which concludes the proof.

F.4 Experimental details

Normalization. In all the networks considered in this paper, after each non-linearity ρ, a 2D batch-normalization layer (Ioffe and Szegedy, 2015) without learned affine parameters sets the per-channel mean and variance across space and data samples to 0 and 1 respectively. After training, we multiply the learned standard deviations by $1/\sqrt{d_j}$ and the learned weight matrices L_{j+1} by $\sqrt{d_j}$ as per our normalization conventions. This ensures that $\mathbb{E}_x[\hat{\phi}_j(x)] = 0$ and $\mathbb{E}_x[\|\hat{\phi}_j(x)\|^2] = 1$, which enables more direct comparisons between networks of different sizes. When evaluating activation convergence for ResNet-18, we explicitly compute these expectations on the training set and standardize the activations $\hat{\phi}_j(x)$ after training for additional numerical stability. When sampling weights from the Gaussian rainbow model, the mean and variance parameters of the normalization layers are computed on the training set before alignment and sampling of the next layer.

Scattering networks. We use the learned scattering architecture of Chapter 6, with several simplifications based on the setting.

The prior operator P_j performs a convolution of every channel of its input with predefined filters: one real low-pass Gabor filter ϕ (a Gaussian window) and 4 oriented Morlet wavelets ψ_θ (complex exponentials localized with a Gaussian window). P_j also implements a subsampling by a factor 2 on even layer indices j, with a slight modification of the filters to compute wavelet coefficients at intermediate scales. See Appendix E.5 for a precise definition of the filters. The learned weight matrices L_j are real for CIFAR-10 experiments, and complex for ImageNet experiments.

We impose a commutation property between P_j and L_j, so that we implement $W_j = P_j L_j$. It is equivalent to having $W_j = L_j P_j$, with the constraint that L_j is applied pointwise with respect to the channels created by P_j. The non-linearity ρ is a complex modulus, which is only applied on the high-frequency channels. A scattering layer writes:

$$\rho W_j z = \left(L_j z * \phi, |L_j z * \psi_\theta|\right)_\theta.$$

The input (and therefore output) of L_j are then both real when L_j is real.

We apply a pre-processing ρP_0 to the input x before feeding it to the network. The fully-connected classifier θ is preceded with a learned 1×1 convolution L_{J+1} which reduces the channel dimension. The learned scattering architecture thus writes:

$$\hat{f}(x) = \theta^{\mathrm{T}} L_{J+1}\, \rho P_J L_J \cdots \rho P_1 L_1\, \rho P_0 x.$$

The number of output channels of L_j is given in Table F.1.

As explained above, we include a 2D batch-normalization layer without learned affine parameters after each non-linearity ρ, as well as before the classifier θ. Furthermore, after each operator L_j, a divisive normalization sets the norm along channels at each spatial location to 1 (except in Figures 7.4, 7.5 and 7.10). There are no learned biases in the architecture beyond the unsupervised channel means.

The non-linearity ρ includes a skip-connection in Figures 7.5 and 7.9, in which case a scattering layer computes

$$\rho W_j z = \left(L_j z * \phi, L_j z * \psi_\theta, |L_j z * \phi|, |L_j z * \psi_\theta|\right)_\theta.$$

In this case, the activations $\phi_j(x)$ are complex. The rainbow model extends to this case by adding complex conjugates at appropriate places. For instance, the alignment matrices become complex unitary operators when both activations and weights are complex.

	j	1	2	3	4	5	6	7	8	9	10	11
CIFAR-10 ($J = 3$)	d_j	64	128	256	512	-	-	-	-	-	-	-
CIFAR-10 ($J = 7$)	d_j	64	128	256	512	512	512	512	512	-	-	-
ImageNet ($J = 10$)	d_j	32	64	64	128	256	512	512	512	512	512	256

Table F.1: Number d_j of output channels of L_j, $1 \leq j \leq J+1$. The total number of projectors is $J+1 = 4$ or $J+1 = 8$ for CIFAR-10 and $J+1 = 11$ for ImageNet.

ResNet. P_j is the patch-extraction operator defined in Section 7.2.3. The non-linearity ρ is a ReLU. We have trained a slightly different ResNet with no bias parameters. In addition, the batch-normalization layers have no learned affine parameters, and are placed after the non-linearity to be consistent with our normalization conventions. The top-5 test accuracy on ImageNet remains at 89% like the original model.

Training. Network weights are initialized with i.i.d. samples from an uniform distribution (Glorot and Bengio, 2010) with so-called Kaiming variance scaling (He et al., 2015), which is the default in the PyTorch library (Paszke et al., 2019). Despite the uniform initialization, weight marginals become Gaussian after a single training epoch. Scattering networks are trained for 150 epochs with an initial learning rate of 0.01 which is divided by 10 every 50 epochs, with a batch size of 128. ResNets are trained for 90 epochs with an initial learning rate of 0.1 which is divided by 10 every 30 epochs, with a batch size of 256. We use the optimizer SGD with a momentum of 0.9 and a weight decay of 10^{-4} (except for Figures 7.4 and 7.10 where weight decay has been disabled). We use classical data augmentations: horizontal flips and random crops for CIFAR, random resized crops of size 224 and horizontal flips for ImageNet. The classification error on the ImageNet validation set is computed on a single center crop of size 224.

Activation covariances. The covariance of the activations $\hat{\phi}_j(x)$ is computed over channels and averaged across space. Precisely, we compute

$$\mathbb{E}_x\left[\sum_u \hat{\phi}_j(x)[u]\, \hat{\phi}_j(x)[u]^{\mathrm{T}}\right],$$

where $\hat{\phi}_j(x)[u]$ is a channel vector of dimension d'_j at spatial location u. It yields a matrix of dimension $d'_j \times d'_j$. For scattering networks, the d'_j channels correspond to the d_j output channels of L_j times the 5 scattering channels computed by P_j (times 2 when ρ includes a skip-connection). For ResNet, $\hat{\phi}_j(x)[u]$ is a patch of size $s_j \times s_j$ centered at u due to the operator P_j. d_j is thus equal to the number d'_j of channels of $\hat{\phi}_j$ multiplied by s_j^2.

Bibliography

Bibliography

Kumar K Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake Richards. α-ReQ : Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. In *Advances in Neural Information Processing Systems*, volume 35, pages 17626–17638, 2022. 118

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022. 102, 108

Jason M Altschuler and Sinho Chewi. Faster high-accuracy log-concave sampling via algorithmic warm starts. *arXiv preprint arXiv:2302.10249*, 2023. 9, 11, 37

Joakim Andén, Vincent Lostanlen, and Stéphane Mallat. Joint time-frequency scattering for audio classification. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2015. 22

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 12

Mathieu Andreux, Tomás Angles, Georgios Exarchakisgeo, Robertozzi Leonardu, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen, Matthew J Hirn, Edouard Oyallon, Sixin Zhang, Carmine E Cella, and Michael Eickenberg. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020. 156, 162

Fabio Anselmi, Lorenzo Rosasco, Cheston Tan, and Tomaso Poggio. Deep convolutional networks are hierarchical kernel machines. *arXiv preprint arXiv:1508.01084*, 2015. 8, 23, 26, 111

Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On invariance and selectivity in representation learning. *Information and Inference: A Journal of the IMA*, 5(2):134–158, 2016. 8

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950. 23, 101

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. 24, 125

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017a. 8, 9, 75

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017b. 9, 23, 103

Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005. 114

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348. Springer, 2014. 11, 36

Krishna Balasubramanian, Sinho Chewi, Murat A Erdogdu, Adil Salim, and Shunshi Zhang. Towards a theory of non-log-concave sampling: first-order stationarity guarantees for langevin monte carlo. In *Conference on Learning Theory*, pages 2896–2923. PMLR, 2022. 32

Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, pages 2269–2277. PMLR, 2021. 24, 125

Matthias Bartelmann and Peter Schneider. Weak gravitational lensing. *Physics Reports*, 340: 291–472, 2001. ISSN 0370-1573. 15, 43

Francesca Bartolucci, Ernesto De Vito, Lorenzo Rosasco, and Stefano Vigogna. Understanding neural networks with reproducing kernel banach spaces. *arXiv preprint arXiv:2109.09710*, 2021. 9

Sergey Bartunov, Adam Santoro, Blake Richards, Luke Marris, Geoffrey E Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Advances in Neural Information Processing Systems*, 31, 2018. 110

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. 123

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. 18

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012. 119

Frederik Benzing, Simon Schug, Robert Meier, Johannes Von Oswald, Yassir Akram, Nicolas Zucchet, Laurence Aitchison, and Angelika Steger. Random initialisations performing above chance and how to find them. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022. 102, 108

Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. 102, 103, 166, 169, 170

Alberto Bietti. *Foundations of deep convolutional models through kernel methods*. Theses, Université Grenoble Alpes, 2019. 26, 98, 105, 111

Alberto Bietti and Francis Bach. Deep equals shallow for ReLU networks in kernel regimes. In *International Conference on Learning Representations*, 2021. 23, 105, 111

Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research*, 20(25):1–49, 2019. 23, 111

Alberto Bietti, Luca Venturi, and Joan Bruna. On the sample complexity of learning under geometric stability. *Advances in neural information processing systems*, 34:18673–18684, 2021. 8

Adam Block, Youssef Mroueh, Alexander Rakhlin, and Jerret Ross. Fast mixing of multi-scale langevin dynamics under the manifold hypothesis. *arXiv preprint arXiv:2006.11166*, 2020. 32

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, 2013. 19, 21, 72, 77, 78, 79, 86, 87, 92

Robert W Buccigrossi and Eero P Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans Image Processing*, 8(12):1688–1701, Dec 1999. 58

Peter J Burt and Edward H Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans Comm*, COM-31(4):532–540, Apr 1983. 8, 58

Emmanuel J Candes, Justin K Romberg, and Terrence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006. 90

Emmanuel Jean Candès, David Leigh Donoho, et al. *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*. Department of Statistics, Stanford University USA, 1999. 9

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007. 9, 167

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188*, 2023. 13, 32

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021. 48

Paul M Chaikin, Tom C Lubensky, and Thomas A Witten. *Principles of condensed matter physics*, volume 10. Cambridge university press, 1995. 38, 42

Antonin Chambolle, Ronald A DeVore, Nam-Yong Lee, and Bradley J Lucier. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans Image Processing*, 7:319–335, Mar 1998. 58

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. *arXiv preprint arXiv:2211.01916*, 2022a. 12, 32, 57

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *International Conference on Learning Representations*, 2021. 46

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001. 9, 18

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022b. 12, 32, 45

Tianshui Chen, Liang Lin, Wangmeng Zuo, Xiaonan Luo, and Lei Zhang. Learning a wavelet-like auto-encoder to accelerate deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 52, 58

Zhengdao Chen, Eric Vanden-Eijnden, and Joan Bruna. A functional-space mean-field theory of partially-trained three-layer neural networks. *arXiv preprint arXiv:2210.16286*, 2022c. 24, 98, 108

Sihao Cheng and Brice Ménard. Weak lensing scattering transform: dark energy and neutrino mass sensitivity. *Monthly Notices of the Royal Astronomical Society*, 507(1):1012–1020, 07 2021. ISSN 0035-8711. 43

Sinho Chewi. *Log-Concave Sampling*. draft, 2023. 9, 11, 32

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in Neural Information Processing Systems*, 31, 2018. 24, 98, 101, 108

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020. 9, 24, 98

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019. 23, 98, 100

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in Neural Information Processing Systems*, 22, 2009. 23, 26, 98, 105

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 110

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 854–863. PMLR, 2017. 72, 76

Peter Clifford and John M Hammersley. Markov fields on finite graphs and lattices. *Unpublished Manuscript*, 1971. 7, 58, 59, 60

Regev Cohen, Yochai Blau, Daniel Freedman, and Ehud Rivlin. It has potential: Gradient-driven denoisers for convergent solutions to inverse problems. *Adv Neural Information Processing Systems (NeurIPS)*, 34, 2021. 58, 61

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2990–2999, 2016. 8, 99, 110

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018. 20

Ronald R Coifman and M Victor Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992. 88

Ronald R Coifman, Yves Meyer, and Victor Wickerhauser. Wavelet analysis and signal processing. In *In Wavelets and their applications*. Citeseer, 1992. 39, 135, 137

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012. 102

F. Cotter and N. G. Kingsbury. A learnable scatternet: Locally invariant convolutional layers. In *2019 IEEE International Conference on Image Processing, ICIP*, pages 350–354. IEEE, 2019. 88

Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. *Advances in Neural Information Processing Systems*, 14, 2001. 102

Matthew S Crouse, Robert D Nowak, and Richard G Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Trans Signal Processing*, 46(4):886–902, 1998. 58

Yan-Qiu Cui and Ke Wang. Markov random field modeling in the wavelet domain for image denoising. In *IEEE Int'l Conf Machine Learning and Cybernetics*, volume 9, pages 5382–5387, 2005. 58, 59

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *Advances in Neural Information Processing Systems*, 29, 2016. 23, 98, 105, 108, 111

Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions. *arXiv preprint arxiv:2209.05442*, Sep 2022. 58

Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992. 136, 141

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11): 1413–1457, 2004. 18

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34, 2021. 48, 49

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *Advances in Neural Information Processing Systems*, 26, 2013. 24, 118

Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in Neural Information Processing Systems*, 27, 2014. 24, 118

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GAN on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 7, 15, 28, 44, 46, 52, 61

Roland L'vovich Dobrushin. The description of the random field by its conditional distributions and its regularity conditions. *Teoriya Veroyatnostei i ee Primeneniya*, 13(2):201–229, 1968. 59

Katharina Dobs, Julio Martinez, Alexander JE Kell, and Nancy Kanwisher. Brain-like functional specialization emerges spontaneously in deep neural networks. *Science advances*, 8(11): eabl8913, 2022. 123

Carles Domingo-Enrich, Alberto Bietti, Eric Vanden-Eijnden, and Joan Bruna. On energy-based models with overparametrized shallow neural networks. In *International Conference on Machine Learning*, pages 2771–2782. PMLR, 2021. 32

David Donoho. Denoising by soft-thresholding. *IEEE Trans Information Theory*, 43:613–627, 1995. 19

David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 90

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences*, 100(5): 2197–2202, 2003. 18

David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 09 1994. 74, 75, 92

David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *Annals of statistics*, 46(4):1742, 2018. 115

Matthias Dorfer, Rainer Kelz, and Gerhard Widmer. Deep linear discriminant analysis. *arXiv preprint arXiv:1511.04707*, 2015. 73

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 6, 88

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, pages 1309–1318. PMLR, 2018. 108

Alexis Dubreuil, Adrian Valente, Manuel Beiran, Francesca Mastrogiuseppe, and Srdjan Ostojic. The role of population structure in computations through neural dynamics. *Nature neuroscience*, 25(6):783–794, 2022. 123

Weinan E and Stephan Wojtowytsch. On the banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *arXiv preprint arXiv:2007.15623*, 2020. 24, 98, 108

Noureddine El Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics*, pages 2757–2790, 2008a. 114

Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, 36(6):2717–2756, 2008b. 115

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 9

Eric Elmoznino and Michael F Bonner. High-performing neural network models of visual cortex benefit from high latent dimensionality. *bioRxiv*, pages 2022–07, 2022. 118

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022. 102, 108

Zhou Fan and Zhichao Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. *Advances in neural information processing systems*, 33:7710–7721, 2020. 23

Kirsten Fischer, Alexandre René, Christian Keup, Moritz Layer, David Dahmen, and Moritz Helias. Decomposing neural networks as mappings of correlation functions. *Physical Review Research*, 4(4):043143, 2022. 24

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936. 17, 72

Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020. 24, 125

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 125

C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations*, 2017. 108

Rinon Gal, Dana Cohen Hochberg, Amit Bermano, and Daniel Cohen-Or. SWAGAN: A style-based wavelet-driven generative model. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 32, 52, 58

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 31, 2018. 108

Leszek Gawarecki and Vidyadhar Mandrekar. Stochastic differential equations. *Stochastic Differential Equations in Infinite Dimensions: with Applications to Stochastic Partial Differential Equations*, pages 73–149, 2011. 105

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020. 24, 98

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distibutions, and bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 6:721–741, Nov 1984. 7, 13, 58, 59

Raja Giryes, Guillermo Sapiro, and Alex M Bronstein. Deep neural networks with random Gaussian weights: A universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13):3444–3457, 2016. 90

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 175

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010. 18, 19

Leonard Gross. Logarithmic sobolev inequalities. *American Journal of Mathematics*, 97(4): 1061–1083, 1975. 11

Arushi Gupta, José Manuel Zorrilla Matilla, Daniel Hsu, and Zoltán Haiman. Non-Gaussian information from weak lensing data via deep learning. *Physical Review D*, 97(10):103515, 2018. 43, 140

Florentin Guth, Simon Coste, Valentin De Bortoli, and Stéphane Mallat. Wavelet score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2022a. 28, 45, 47, 49, 50, 53

Florentin Guth, John Zarka, and Stéphane Mallat. Phase Collapse in Neural Networks. In *International Conference on Learning Representations*, 2022b. 28

Florentin Guth, Etienne Lempereur, Joan Bruna, and Stéphane Mallat. Conditionally strongly log-concave generative models. In *International Conference on Machine Learning*, 2023a. 28, 31, 33, 37

Florentin Guth, Brice Ménard, Gaspar Rochette, and Stéphane Mallat. A rainbow in deep network black boxes. *arXiv preprint arXiv:2305.18512*, 2023b. 28

Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Math Annal.*, 69:331–371, 1910. 62

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing Systems*, 5, 1992. 116

Trevor Hastie and Robert Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987. 167

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. 85

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986. 48

James V Haxby, J Swaroop Guntupalli, Andrew C Connolly, Yaroslav O Halchenko, Bryan R Conroy, M Ida Gobbini, Michael Hanke, and Peter J Ramadge. A common, high-dimensional model of the representational space in human ventral temporal cortex. *Neuron*, 72(2):404–416, 2011. 102

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 175

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 17, 20, 72, 79, 88, 112

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. 55

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 6

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. 7, 44, 46, 47, 52, 54, 55, 57, 61, 64, 67

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 15, 28, 44, 46, 51, 52, 58, 60, 61

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 113

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pages 9–1. JMLR Workshop and Conference Proceedings, 2012. 37

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 125

John R Hurley and Raymond B Cattell. The Procrustes program: Producing direct rotation to test a hypothesized factor structure. *Behavioral science*, 7(2):258, 1962. 102

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 9, 35, 46, 48

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, page 448–456, 2015. 5, 73, 162, 174

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018. 23, 98

Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *International Conference on Machine Learning*, pages 4631–4640. PMLR, 2020. 9

Stéphane Jaffard. Wavelelt methods for fast resolution elliptic problems. *SIAM Journal on Numerical Analysis*, 29(5):965–986, 1992. 53

Hicham Janati, Boris Muzellec, Gabriel Peyré, and Marco Cuturi. Entropic optimal transport between unbalanced Gaussian measures has a closed form. *Advances in Neural Information Processing Systems*, 33:10468–10479, 2020. 170

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009. 22, 98, 100

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. 125

Yuling Jiao, Bangti Jin, and Xiliang Lu. Iterative soft/hard thresholding with homotopy continuation for sparse recovery. *IEEE Signal Processing Letters*, 24(6):784–788, 2017. 18

Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models, 2022. 46, 52

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of statistics*, 29(2):295–327, 2001. 114

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021. 46, 52, 55

Zahra Kadkhodaie and Eero P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Adv Neural Information Processing Systems (NeurIPS*21)*, volume 34, Dec 2021. 13, 46, 55, 58, 61, 64, 152, 153

Zahra Kadkhodaie, Florentin Guth, Stéphane Mallat, and Eero P Simoncelli. Learning multi-scale local conditional probability models of images. In *International Conference on Learning Representations*, volume 11, 2023. 28, 57

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 113

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. arxiv 2017. *arXiv preprint arXiv:1710.10196*, pages 1–26, 2018. 47, 53, 65, 152

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022. 13

Jevgenjis. Kaupužs, Roderick VN Melnik, and J Rimšāns. Corrections to finite-size scaling in the φ^4 model on square lattices. *International Journal of Modern Physics C*, 27(09):1650108, 2016. 38

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Adv Neural Information Processing Systems (NeurIPS)*, 34:21757–21769, 2021. 58, 61

Martin Kilbinger. Cosmology with cosmic shear observations: a review. *Reports on Progress in Physics*, 78(8):086901, jul 2015. 15, 43

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5, 149

Diederik P Kingma and Ruiqi Gao. Understanding the diffusion objective as a weighted integral of elbos. *arXiv preprint arXiv:2303.00848*, 2023. 13

Frederic Koehler, Alexander Heckett, and Andrej Risteski. Statistical efficiency of score matching: The view from isoperimetry. *arXiv preprint arXiv:2210.00726*, 2022. 9, 11, 32, 36

Andrey N Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962. 50

Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2747–2755, 2018. 8, 99, 110

Risi Kondor, Zhen Lin, and Shubendhu Trivedi. Clebsch–Gordan nets: a fully Fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems 31*, pages 10138–10147, 2018. 20

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021. 46, 52, 55

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations*, 2021. 46

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 24, 28, 98, 99, 102, 108, 113

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 76, 79, 87, 112

Alex Krizhevsky. Convolutional deep belief networks on cifar-10. Technical report, University of Toronto, 2010. 77

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012. 6, 17, 19, 77, 86, 87, 123, 162

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001. 7

Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: when Langevin meets Tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022. 61

Rene Laureijs, J Amiaux, S Arduini, J-L Augueres, J Brinchmann, R Cole, M Cropper, C Dabin, L Duvet, A Ealet, et al. Euclid definition study report. *arXiv preprint arXiv:1110.3193*, 2011. 33, 43

Erwan Le Pennec and Stéphane Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4):423–438, 2005. 9

Guillaume Leclerc and Aleksander Madry. The two regimes of deep network training. *arXiv preprint arXiv:2002.10376*, 2020. 125

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 6, 109

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2, 1989a. 5, 6, 109

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2, 1989b. 116

Yann LeCun, Corinna Cortes, and Chris J Burges. MNIST handwritten digit database. *ATT Labs [Online]*, 2, 2010. URL http://yann.lecun.com/exdb/mnist. 76, 87

Yann LeCun, Yoshu Bengio, and Geoffrey E Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1, 5, 72

Gregory R Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36): 1237, 2019a. 141

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018. 23, 98, 108

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in Neural Information Processing Systems*, 32, 2019b. 23, 98

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020. 24, 98

Shuo-Hui Li. Learning non-linear wavelet transformation via normalizing flow. *arXiv preprint arXiv:2101.11306*, 2021. 52, 58

Yundong Li, Weigang Zhao, and Jiahao Pan. Deformable patterned fabric defect detection with Fisher criterion-based deep learning. *IEEE Transactions on Automation Science and Engineering*, 14(2):1256–1264, 2016. 73

Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019. 112, 122

Zhiyuan Li, Yi Zhang, and Sanjeev Arora. Why are convolutional nets more sample-efficient than fully-connected nets? In *International Conference on Learning Representations*, 2021. 8

Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations (ICLR)*, 2019. 18

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds, 2022a. 46, 52, 55

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022b. 6

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, December 2015. 54, 62, 152

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 46, 52, 55

Siwei Lyu and Eero P Simoncelli. Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures. *IEEE Trans Pattern Analysis and Machine Intelligence*, 31 (4):693–706, Apr 2009. 58

Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013. 115

Shahin Mahdizadehaghdam, Ashkan Panahi, Hamid Krim, and Liyi Dai. Deep dictionary learning: A parametric network approach. *IEEE Transactions on Image Processing*, 28(10):4790–4802, Oct 2019. 19, 84, 89, 91, 92

Julien Mairal. End-to-end kernel learning with supervised convolutional kernel networks. *Advances in Neural Information Processing Systems*, 29, 2016. 23, 26, 98, 105

Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2009. 18

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011. 18

Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. *Advances in Neural Information Processing Systems*, 27, 2014. 23, 111, 112, 122

Maurits Malfait and Dirk Roose. Wavelet-based image denoising using a Markov random field a priori model. *IEEE Trans Image Processing*, 6(4):549–565, 1997. 58, 59

Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:674–693, 1989. 50, 135, 146, 147

Stéphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008. 8, 9, 58, 59, 78, 136, 137, 148

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 8, 19, 20, 72, 80, 83, 87, 111

Stéphane Mallat. Understanding deep convolutional networks. *Phil. Trans. of Royal Society A*, 374(2065), 2016. 22

Stéphane Mallat, Sixin Zhang, and Gaspar Rochette. Phase harmonic correlations and convolutional neural networks. *Information and Inference: A Journal of the IMA*, 9(3):721–747, 11 2019. 21, 78, 89

Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022. 113

Benoît B Mandelbrot. The fractal geometry of nature/revised and enlarged edition. *New York*, 1983. 50

Vladimir A Marčenko and Leonid A Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967. 121

Tanguy Marchand, Misaki Ozawa, Giulio Biroli, and Stéphane Mallat. Wavelet conditional renormalization group. *arXiv preprint arXiv:2207.04941*, 2022. 10, 14, 15, 28, 32, 38, 39, 40, 43, 45, 46, 51, 57, 58, 59, 60, 142, 143

Peter A Markowich and Cédric Villani. On the trend to equilibrium for the Fokker-Planck equation: an interplay between physics and functional analysis. *Mat. Contemp*, 19:1–29, 2000. 11

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *The Journal of Machine Learning Research*, 22(1):7479–7551, 2021. 24, 98, 105, 115, 121

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018. 23, 98, 108

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layers neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018. 24, 98, 101, 108

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory*, pages 3351–3418. PMLR, 2021. 8

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022. 9, 23

Yves Meyer. *Wavelets and Operators*. Advanced mathematics. Cambridge university press, 1992. 52, 53

M Kivanc Mihçak, Igor Kozintsev, Kannan Ramchandran, and Pierre Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Trans Signal Processing*, 6(12):300–303, Dec 1999. 58

Stanislav Minsker. On some extensions of Bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119, 2017. 103

Koichi Miyasawa. An empirical Bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist.*, 38:181–188, 1961. 61, 152

Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *International Conference on Learning Representations*, 2019. 63, 75, 87, 100, 152

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018. 102

Jaouad Mourtada. Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. *The Annals of Statistics*, 50(4):2157–2178, 2022. 37

Eliya Nachmani, Robin San Roman, and Lior Wolf. Non Gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021. 46, 52, 55

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 5

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 1996. 23, 98, 108

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016. 6

Phan-Minh Nguyen and Huy Tuan Pham. A rigorous framework for the mean field limit of multilayer neural networks. *arXiv preprint arXiv:2001.11443*, 2020. 24, 98, 108

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021. 13, 46, 50, 54, 55, 149

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. 123

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 9

Jordan Ott, Erik Linstead, Nicholas LaHaye, and Pierre Baldi. Learning in the machine: To share or not to share? *Neural Networks*, 126:235–249, 2020. ISSN 0893-6080. 110

Edouard Oyallon. *Analyzing and Introducing Structures in Deep Convolutional Neural Networks*. Theses, Paris Sciences et Lettres, 2017. 83

Edouard Oyallon. Building a regular decision boundary with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1886–1894, 2017. 17, 22, 72

Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2865–2873. IEEE Computer Society, 2015. 22, 72, 80, 112, 122

Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5618–5627, 2017. 22, 81

Rupert Paget and I Dennis Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE transactions on image processing*, 7(6):925–931, 1998. 58

Biraj Pandey, Marius Pachitariu, Bingni W Brunton, and Kameron Decker Harris. Structured random receptive fields enable informative sensory encodings. *PLoS Computational Biology*, 18(10):e1010484, 2022. 122

Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet Hessians. In *International Conference on Machine Learning*, pages 5012–5021. PMLR, 2019. 116

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra, 2020. 17, 72

Vardan Papyan, X Y Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 2020. 17, 72, 83

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 175

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019. 103

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195, 1999. 75

Nicolas Pinto, David Doukhan, James J DiCarlo, and David D Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579, 2009. 22, 98, 100

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017. 8

Roman Pogodin, Yash Mehta, Timothy Lillicrap, and Peter E Latham. Towards biologically plausible convolutional networks. *Advances in Neural Information Processing Systems*, 34: 13924–13936, 2021. 110

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016. 23

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiV:2105.06337*, 2021. 46

Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans Image Processing*, 12 (11):1338–1351, Nov 2003. 58

Matthew B Priestley. Evolutionary spectra and non-stationary processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 27(2):204–229, 1965. 86

Qiang Qiu, Xiuyuan Cheng, Robert Calderbank, and Guillermo Sapiro. DCFNet: Deep neural network with decomposed convolutional filters. *International Conference on Machine Learning*, 2018. 80, 88

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, 30, 2017. 24, 98, 99, 102, 108, 113, 119

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007. 22, 23, 98, 100, 101, 118

Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *46th annual Allerton conference on communication, control, and computing*, pages 555–561. IEEE, 2008. 9, 23, 101

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv e-prints*, art. arXiv:2204.06125, April 2022. 7, 32, 60, 66, 67

C R Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society: Series B (Methodological)*, 10(2):159–193, 1948. 17, 72

M Raphan and E P Simoncelli. Learning to be Bayesian without supervision. In *Adv Neural Information Processing Systems*, volume 19, May 2007. 152

Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019. 118

Markus Riedle. Cylindrical wiener processes. *Séminaire de Probabilités XLIII*, pages 191–214, 2011. 105

H Robbins. An empirical bayes approach to statistics. In *Proc Third Berkeley Symposium on Mathematical Statistics and Probability*, volume I, pages 157–163. University of CA Press, 1956. 152

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 7, 32, 60, 66, 67

O Ronneberger, P Fischer, and T Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int'l Conf Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 6

Grant M Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*, 2018. 24, 98, 101, 108

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. *Advances in neural information processing systems*, 30, 2017. 9, 23

Alessandro Rudi, Guillermo D Canas, and Lorenzo Rosasco. On the sample complexity of subspace learning. In *Advances in Neural Information Processing Systems*, volume 26, 2013. 103

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6, 19, 79, 86, 112, 123

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the Hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016. 116

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017. 116

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021. 15, 28, 44, 46, 51, 52, 54, 149

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 7, 32, 60, 66, 67

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 46, 52, 55

Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021. 46, 52, 55

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. 125

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019. 125

Meyer Scetbon and Zaid Harchaoui. A spectral analysis of dot-product kernels. In *International conference on artificial intelligence and statistics*, pages 3394–3402. PMLR, 2021. 23, 113, 118

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017. 23

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 8, 101

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks—ICANN'97: 7th International Conference Lausanne, Switzerland, October 8–10, 1997 Proceeedings*, pages 583–588. Springer Verlag, 1997. 101

Peter H Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 102

Dominik Schröder, Hugo Cui, Daniil Dmitriev, and Bruno Loureiro. Deterministic equivalent and error universality of deep random features learning. In *International Conference on Machine Learning*, 2023. 23

Friedrich Schuessler, Francesca Mastrogiuseppe, Alexis Dubreuil, Srdjan Ostojic, and Omri Barak. The interplay between randomness and structure during learning in RNNs. *Advances in Neural Information Processing Systems*, 33:13352–13362, 2020. 125

Ivan W Selesnick, Richard G Baraniuk, and Nick C Kingsbury. The dual-tree complex wavelet transform. *IEEE signal processing magazine*, 22(6):123–151, 2005. 78

Levent Şendur and Ivan W Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans Signal Processing*, 50(11):2744–2756, Nov 2002. 58

Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some CNNs. *Nature Communications*, 14(1):908, 2023. 24

James P Sethna. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*, volume 14. Oxford University Press, USA, 2021. 38, 42

Haozhe Shan and Blake Bordelon. A theory of neural tangent kernel alignment and its influence on training. *arXiv preprint arXiv:2105.14301*, 2021. 24

Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *international conference on machine learning*, pages 2217–2225. PMLR, 2016. 86, 87

Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Jonathan Ragan-Kelley, Ludwig Schmidt, and Benjamin Recht. Neural kernels without tangents. In *International Conference on Machine Learning*, pages 8614–8623. PMLR, 2020. 122

David Sherrington and Scott Kirkpatrick. Solvable Model of a Spin-Glass. *Phys. Rev. Lett.*, 35 (26):1792–1796, dec 1975. 59

Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1233–1240, 2013. 22, 92, 110

Patrice Y Simard, David Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, 2003. 77

Eero P Simoncelli and William T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proceedings., International Conference on Image Processing*, volume 3, pages 444–447. IEEE, 1995. 78

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 6, 17

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020. 24, 98, 101, 108

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of deep neural networks. *Mathematics of Operations Research*, 47(1):120–152, 2022. 24, 98, 108

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. 5

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*, 2017. 102

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proc 32nd Int'l Conf on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. 7, 57

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. *arXiv preprint arXiv:2212.03860*, 2022. 13, 32

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 46, 52, 55

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 7, 46, 47, 57, 64

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020. 50

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34: 1415–1428, 2021a. 12, 13

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. 7, 44, 46, 47, 57, 60, 61, 64

Bharath K Sriperumbudur and Nicholas Sterge. Approximate kernel PCA: Computational versus statistical trade-off. *The Annals of Statistics*, 50(5):2713–2736, 2022. 103, 113

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Aapo Hyvärinen, and Revant Kumar. Density estimation in infinite dimensional exponential families. *arXiv preprint arXiv:1312.3516*, 2013. 32, 37

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571 (7765):361–365, 2019. 118

Andre Stuhlsatz, Jens Lippel, and Thomas Zielke. Feature extraction with deep neural networks by a generalized discriminant analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):596–608, 2012. 73

Jeremias Sulam, Vardan Papyan, Yaniv Romano, and Michael Elad. Multilayer convolutional sparse modeling: Pursuit and dictionary learning. *IEEE Transactions on Signal Processing*, 66(15):4090–4104, 2018. 19, 84, 89, 91, 92

Jeremias Sulam, Aviad Aberdam, Amir Beck, and Michael Elad. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 19, 84, 89

Nazneen N Sultana, Bappaditya Mandal, and Niladri B Puhan. Deep residual network with regularised Fisher framework for detection of melanoma. *IET Computer Vision*, 12(8):1096–1104, 2018. 73

Kai Sun, Jiangshe Zhang, Hongwei Yong, and Junmin Liu. FPCANet: Fisher discrimination for principal component analysis network. *Knowledge-Based Systems*, 166:108–117, 2019. 73

Xiaoxia Sun, Nasser M Nasrabadi, and Trac D Tran. Supervised deep sparse coding networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 346–350, 2018. 19, 84, 89, 91, 92

Danica J Sutherland, Heiko Strathmann, Michael Arbel, and Arthur Gretton. Efficient and principled score estimation with nyström kernel exponential families. In *International Conference on Artificial Intelligence and Statistics*, pages 652–660. PMLR, 2018. 32, 37

Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1), 2019. URL `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`. 6

Matthias Thamm, Max Staats, and Bernd Rosenow. Random matrix analysis of deep neural network weight matrices. *Physical Review E*, 106(5):054124, 2022. 24, 98, 105, 115, 121, 125

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996. 9, 18

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 88

Asher Trockman, Devin Willmott, and J Zico Kolter. Understanding the covariance structure of convolutional filters. In *International Conference on Learning Representations*, 2023. 111

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012. 103

Mark Tygert, Joan Bruna, Soumith Chintala, Yann LeCun, Serkan Piantino, and Arthur Szlam. A mathematical motivation for complex-valued convolutional networks. *Neural computation*, 28(5):815–825, 2016. 86

Matej Ulicny, Vladimir A Krylov, and Rozenn Dahyot. Harmonic networks for image classification. In *Proceedings of the British Machine Vision Conference*, Sep. 2019. 80, 88

Nicholas Vakhania, Vazha Tarieladze, and S Chobanyan. *Probability distributions on Banach spaces*, volume 14. Springer Science & Business Media, 1987. 105

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 13

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012. 37

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. 12, 46, 61

Martin J Wainwright and Eero P Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. *Advances in neural information processing systems*, 12, 1999. 21, 46, 51

Martin J Wainwright, Odelia Schwartz, and Eero P Simoncelli. Natural image statistics and divisive normalization: Modeling nonlinearities and adaptation in cortical neurons. *Statistical Theories of the Brain*, 01 2001a. 93, 162

Martin J Wainwright, Eero P Simoncelli, and Alan S Willsky. Random cascades on wavelet trees and their use in modeling and analyzing natural imagery. *Applied and Computational Harmonic Analysis*, 11(1):89–123, Jul 2001b. 58

Ross Wightman, Hugo Touvron, and Hervé Jégou. ResNet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021. 6

Christopher Williams. Computing with infinite networks. *Advances in Neural Information Processing Systems*, 9, 1996. 23, 98, 108

Kenneth G Wilson. Renormalization group and critical phenomena. II. Phase-space cell analysis of critical behavior. *Physical Review B*, 4(9):3184, 1971. 14, 39, 46, 59

Kenneth G Wilson. The renormalization group and critical phenomena. *Reviews of Modern Physics*, 55(3):583, 1983. 51

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020. 98

Lin Wu, Chunhua Shen, and Anton Van Den Hengel. Deep linear discriminant analysis on Fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 65:238–250, 2017. 73

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv preprint arXiv:2112.07804*, 2021. 46, 52, 55

Greg Yang and Edward J Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021. 23, 24, 98, 100, 108

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022. 113

Zai Yang, Cishen Zhang, and Lihua Xie. On phase transition of compressed sensing in the complex domain. *IEEE Signal Processing Letters*, 19(1):47–50, Jan 2012. ISSN 1558-2361. 89

Jason J Yu, Konstantinos G Derpanis, and Marcus A Brubaker. Wavelet flow: Fast training of high resolution normalizing flows. *Advances in Neural Information Processing Systems*, 33: 6184–6196, 2020. 32, 52, 58

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7370–7379, 2017. 24, 118

John Zarka. *Sparsity and Phase Collapse in Deep Convolutional Networks*. Theses, Paris Sciences et Lettres, 2022. 19, 21, 71, 83

John Zarka, Louis Thiry, Tomas Angles, and Stéphane Mallat. Deep network classification by scattering and homotopy dictionary learning. In *International Conference on Learning Representations*, 2020. 19, 81, 84, 89, 91

John Zarka, Florentin Guth, and Stéphane Mallat. Separation and concentration in deep networks. In *International Conference on Learning Representations*, 2021. 28, 71, 75

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014. 17

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator, 2022. 46, 52, 55

Jean Zinn-Justin. *Quantum Field Theory and Critical Phenomena: Fifth Edition*. Oxford University Press, 04 2021. ISBN 9780198834625. 38, 42

José Manuel Zorrilla Matilla, Zoltán Haiman, Daniel Hsu, Arushi Gupta, and Andrea Petri. Do dark matter halos explain lensing peaks? *Phys. Rev. D*, 94:083506, Oct 2016. 43, 140

RÉSUMÉ

Les réseaux de neurones convolutifs profonds ont obtenu un succès considérable en vision par ordinateur, à la fois pour l'apprentissage non-supervisé (i.e., génération d'image) et l'apprentissage supervisé (i.e., classification d'image). Cependant, les principes fondamentaux derrière ces résultats impressionnants ne sont pas bien compris. En particulier, l'apprentissage profond semble échapper à la malédiction de la dimensionalité, ce qui révèle une structure mathématique riche dans les problèmes d'apprentissage rencontrés en pratique. Cette structure est présente dans les interactions entre les données d'entraînement (sur quelles propriétés se repose-t-on implicitement ?), l'architecture (quel est le rôle fonctionnel rempli par ses composants ?) et l'algorithme d'optimisation (qu'est-ce que le réseau a appris ?). Cette thèse comporte des résultats sur ces trois questions. Premièrement, nous montrons qu'une factorisation multi-échelles des distributions d'images peut révéler des propriétés de régularité, des structures de dépendances markoviennes locales, et même de la log-concavité conditionnelle, alors que la distribution globale ne possède pas ces propriétés. Cela conduit à des algorithmes efficaces d'apprentissage et d'échantillonnage dont on peut contrôler toutes les sources d'erreurs. Deuxièmement, nous étudions le rôle de la non-linéarité en classification d'images, et montrons que sa fonction principale est de collapser la phase complexe des coefficients d'ondelettes des activations du réseau. En revanche, des modèles précédents reposant sur des seuillages et des hypothèses de parcimonie ne sont ni suffisants ni nécessaires pour expliquer la précision de classification des réseaux profonds. Troisièmement, nous introduisons un modèle probabiliste des poids appris dans les architecture profondes, en capturant les dépendances entre couches par un alignement des activations du réseau sur une représentation déterministe associée à un noyau reproduisant. Le modèle est spécifié à travers des distributions à chaque couche, dont les covariances sont de bas rang et réalisent une réduction de dimensionalité entre les plongements en haute dimension calculés par la non-linéarité. Dans certains cas, ces distributions sont approximativement gaussiennes, et leurs covariances capturent la performance et la dynamique d'entraînement du réseau.

MOTS CLÉS

réseaux de neurones convolutifs ⋆ apprentissage profond ⋆ vision par ordinateur ⋆ classification d'images ⋆ génération d'images ⋆ représentations multi-échelles

ABSTRACT

Deep convolutional neural networks have achieved considerable success in computer vision tasks, both in unsupervised learning (e.g., image generation) and supervised learning (e.g., image classification). However, the fundamental principles behind these impressive results remain not well understood. In particular, deep learning seemingly escapes the curse of dimensionality in practice, which evidences a rich mathematical structure underlying real-world learning problems. This structure is revealed by the interplay between the training data (what properties are we implicitly relying on?), the architecture (what is the functional role of network computations?), and the optimization algorithm (what has the network learned?). This thesis presents results on these three questions. First, we demonstrate that a multiscale factorization of image distributions can reveal properties of smoothness, local Markov dependency structure, and even conditional log-concavity, whereas the global distribution does not enjoy these properties. It leads to efficient learning and sampling algorithms where all sources of errors can be controlled. Second, we investigate the role of non-linearity in image classification, and show that its main function is to collapse the phase of complex wavelet coefficients of network activations. In contrast, previous models based on thresholding and sparsity assumptions are neither sufficient nor necessary to explain the classification accuracy of deep networks. Third, we introduce a probabilistic model of learned weights in deep architectures, with layer dependencies that are captured by alignment of the network activations to deterministic kernel embeddings. The model is specified through weight distributions at each layer, whose covariances are low-rank and perform dimensionality reduction in-between the high-dimensional embeddings computed by the non-linearities. In some cases, these weight distributions are approximately Gaussian, and their covariances capture the performance and training dynamics of the network.

KEYWORDS

convolutional neural networks ⋆ deep learning ⋆ computer vision ⋆ image classification ⋆ image generative modeling ⋆ multiscale representations