Natural Image Densities:

Learning, Understanding and Utilizing

by

Zahra Kadkhodaie

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Center for Data Science

New York University

September, 2024

Dr. Eero P. Simoncelli

DEDICATION

To my mother who is the light in my world.

Acknowledgments

First and foremost, I want to thank my advisor, Eero Simoncelli, for his constant and generous support throughout this exciting academic journey, for thoughtfully guiding me in the landscape of science yet giving me the freedom to explore, and for creating a fantastic environment which led to many intellectual adventures and several fruitful collaborations. Eero is a researcher whose depth of knowledge does not constrain his scientific curiosity. He is the mentor I aspire to become.

I would like to extend my gratitude to Stéphane Mallat, whose commitment to intellectual rigor and passion for discovery led us to many exhilarating conversations. The work presented in Chapter 3 and Chapter 4 would not have been possible without his contributions. I also want to thank David Brainard, whose genuine passion for scientific discussions, combined with his generous attitude, sets an example of what it means to be a true scholar.

At the Center for Data Science, I am grateful to my committee members, Joan Bruna, Andrew Gordon Wilson, and Kyunghyun Cho, whose support and scrutiny helped me refine my thinking and improve my work. I am also thankful to Léon Bottou, my external committee member, for his feedback, which led to several thought-provoking questions.

My time at NYU was made much more enjoyable by the amazing group of people I am privileged to call lab mates. I was very lucky to meet and build friendships with Caroline Haimerl, whose passion for life (and dance!) and her determination were always contagious; Florentin Guth, an excellent collaborator with an exquisite sense of humor; Pierre-Étienne Fiquet, whose endless appetite for science and art shaped many conversations within our circle; Lyndon Duong, a kind and always reliable comrade, whether teaming up for a course project or taking a lunch break; Teddy Yerxa, with an incredible talent in both building deep net models and in extracting humor from ordinary events; Nikhil Parthasarathy, my favorite office mate, with whom I had many valuable conversations about the purpose of science, the influence of our work, and, of course, how to debug code; Jenelle Feather, an enthusiastic scientist and potter with a refreshingly inquisitive mind; Billy Broderick who never hesitated to answer my random questions about coding or tea; and David Lipshutz, a brilliant and sincere researcher who quickly became an integral part of our lab. I was also fortunate to work with Lingqi Zhang, whose internship project at the Center for Computational Neuroscience, Flatiron institute, turned into an interesting collaboration. Additionally, I am grateful to Jessica Hauser for her tireless support in creating the best environment any researcher could wish for.

Last but not least, I am deeply grateful to my family for being an unwavering source of love and support. To my mother who is the light in my world and to whom I owe everything. To my father who ingrained in us the value of expertise and encouraged us to be ambitious. To my sister who is a perfect amalgam of brilliance and kindness. To my brother who is a source of countless joyful moments and shared laughter. And I am also particularly thankful to Mehrdad, my friend and companion, who brightened my life by being the person he is and for helping me to grow. This would have not been possible without his constant support.

Abstract

Many problems in image processing and computer vision rely, explicitly or implicitly, on statistical density models. Describing the full density of natural images, $p(x)$, is a daunting problem given the dimensionality of the signal space. Traditionally, models have been developed by combining assumed symmetry properties, with simple parametric forms, often within prespecified transformed coordinate systems. While these models have led to steady advances in problems such as denoising, they are too simplistic to generate complex features that occur in our visual world.

Deep neural networks have provided state-of-the-art solutions for problems such as denoising, which implicitly rely on a prior probability model of natural images. Here, we first develop a robust and general methodology for extracting the prior. We rely on a statistical result due to Tweedie (1956) and Miyasawa (1961), who showed that the least-squares solution for removing additive Gaussian noise can be written directly in terms of the gradient of the log of the noisy signal density. We use this fact to develop a stochastic coarse-to-fine gradient ascent procedure for drawing high-probability samples from the implicit prior embedded within a neural network trained to perform blind (i.e., unknown noise level) least-squares denoising. This algorithm is similar to score-based diffusion framework, yet different in several ways.

Unlike the classical framework, we do not have direct access to the learned density, which gives rise to a crucial question: *what is the prior?* The rest of the thesis focuses on understanding and using this prior.

At the core of our coarse-to-fine gradient ascent sampling algorithm is a deep neural network (DNN) denoiser. Despite their success, we lack an understanding of the DNN denoiser mechanisms and more importantly what priors are being learned by these models. In order to make the DNN denoiser interpretable, we remove all network biases (i.e. additive constants), to enforce the denoising mapping to become locally linear. This architecture lends itself to local linear algebraic analysis through the Jacobian of the denoising map, which provides a high level interpretability. A desired side effect of locally linear models is that they generalize automatically across noise levels.

Next, we study the continuity of the implicit image prior. We design an experiment to investigate whether the prior interpolates between the training examples or consists of a discrete set of delta functions corresponding to a memorized set of training examples. We find that for small datasets, the latter is the case. But with large enough datasets, the network generalizes beyond training examples, evidenced by high quality novel generated samples. Surprisingly, we observe that, for large enough datasets, two models trained on non-overlapping subsets of a dataset learn nearly the same density. We analyze the learned denoising functions and show that the inductive biases give rise to a shrinkage operation in a basis adapted to the underlying image. Examination of these bases reveals oscillating harmonic structures along contours and in homogeneous regions. We demonstrate that trained denoisers are inductively biased towards these geometry-adaptive harmonic bases.

Having established that a DNN denoiser can generalize, we employ the learned image density to study the question of low-dimensionality of image priors. The goal is to exploit image properties to factorize the density into low dimensional densities, thereby reducing the number of parameters and training examples. To this end, we develop a low-dimensional probability model for images decomposed into multi-scale wavelet sub-bands. 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 effective size of Markov neighborhoods (i.e. the size w.r.t to the grid size) grows from fine to coarser scales. 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 evaluate our model and show that locality and stationarity assumptions hold for conditional RF sizes as small as 9×9 without harming performance. Thus, high-dimensional score estimation for images can be reduced to low-dimensional Markov conditional models, alleviating the curse of dimensionality.

Finally, we put the denoiser prior into use. A generalization of the coarse-to-fine gradient ascent sampling algorithm to constrained sampling provides a method for using the implicit prior to solve any linear inverse problem, with no additional training. We demonstrate the generality of the algorithm by using it to produce high-quality solutions in multiple applications, such as deblurring, colorization, compressive sensing, and super resolution.

Contents

Dedication **iii**

Acknowledgments **iv**

Abstract **vi**

List of Figures **xi**

List of Tables **xv**

List of Appendices **xvi**

1 Introduction **1**

1.1 Classical image density models 2

1.2 Image density models in the era of deep learning 6

2 Learning image densities from data using a denoiser **14**

2.1 Image prior implicit in a denoiser 14

2.2 Drawing high-probability samples from the implicit prior 18

2.3 Deep Neural Network Denoisers 28

2.4 Discussion 41

3 Generalization beyond memorizing training images **43**

3.1 Diffusion model variance and denoising generalization 45

3.2 Inductive biases 57

3.3 Discussion 67

4 Muli-scale Local conditional image density 69

4.1 Theory of Markrov random fields 71

4.2 MRF image models for solving inverse problems 74

4.3 Markov Wavelet Conditional Models 84

4.4 Score-based Markov Wavelet Conditional Models 86

4.5 Markov Wavelet Conditional Denoising 88

4.6 Markov Wavelet Conditional Super-resolution and Synthesis 93

4.7 Discussion 97

5 Using learned image densities to solve linear inverse problems 99

5.1 Constrained sampling algorithm 100

5.2 Linear inverse examples 104

5.3 Optimal linear measurements 119

5.4 Related work 121

5.5 Discussion 124

6 Discussion 126

Appendices 129

Bibliography 154

List of Figures

1.1 Position-frequency partitions of different representations 4

2.1 Two-dimensional simulation/visualization of the sampler 19

2.2 Convergence of SISA sampling algorithm 22

2.3 Visualization of SISA sampling algorithm trajectories 23

2.4 Visualization of SISA sampling algorithm trajectories for color images 23

2.5 Visualization of SISA sampling algorithm trajectories for MNIST 23

2.6 The effect of hyper-parameter β on generated samples of natural patches 24

2.7 The effect of hyper-parameter β on generated samples on MNIST digits 24

2.8 Generalized conic manifold 29

2.9 Bias component of the locally affine denoising mapping 31

2.10 Effect of net bias on generalization for an example image. 31

2.11 Effect of net bias on generalization for a test dataset 32

2.12 Visualization of the linear weighting functions of a BF-CNN 35

2.13 Analysis of the SVD of the Jacobian of a BF-CNN for ten natural images 37

2.14 Visualization of left singular vectors of the Jacobian of a BF-CNN 38

2.15 Signal subspace properties. 40

3.1 Memorization vs. generalization in 2D 44

3.2 Transition from memorization to generalization 48

3.3 Convergence of model variance. 50

3.4 Similarity between data subsets 51

3.5 More examples to illustrate convergence of model variance 51

3.6 Bifurcation of trajectories. 51

3.7 Transition from memorization to generalization, for a UNet denoiser trained on bedroom LSUN images 53

3.8 Transition from memorization to generalization for a BF-CNN denoiser 54

3.9 Convergence of model variance for a BF-CNN denoiser. 54

3.10 Convergence of model variance on LSUN bedroom dataset 55

3.11 Convergence for LSUN dataset 55

3.12 Generalization scaling law 56

3.13 Analysis of a denoiser trained on face images 62

3.14 Analysis of denoiser trained on $\mathbf{C}^\alpha$ images 64

3.15 Analysis of a BF-CNN denoiser trained on a single face image 65

3.16 Analysis of a BF-CNN denoiser is trained on a set of 2D sine wave images 65

3.17 Analysis of a denoiser trained on a disk dataset 66

3.18 Analysis of a denoiser trained on a dataset of shuffled faces 67

4.1 Markov neighborhoods on a regular lattice 74

4.2 Markov Random Field for restoration 77

4.3 Markov neighborhoods in GSM model 78

4.4 Hierarchical GSM 79

4.5 Markov random field for segmentation 81

4.6 Non-Gaussian pair-wise clique potentials 82

4.7 Markov random field for texture synthesis 83

4.8 Markov wavelet conditional model structure 86

4.9 Wavelet conditional denoiser architecture 88

4.10 Comparison of denoiser performance on pixel vs wavelet domain 91
4.11 Denoising examples. 92
4.12 Super-resolution examples. 95
4.13 Image synthesis. 96

5.1 Two-dimensional simulation/visualization of constrained sampling 100
5.2 Block diagrams for denoiser training, and Universal Inverse Sampler 103
5.3 Spatial super-resolution. 105
5.4 Deblurring (spectral super-resolution) 107
5.5 Compressive sensing 109
5.6 comparison with a deep net method in compressive sensing 110
5.7 Inpainting examples. 111
5.8 Sampling diversity 112
5.9 Recovery of randomly selected missing pixels 113
5.10 Colorization of grayscale images 114
5.11 Reconstruction from cone excitations is robust to cone cell proportions 115
5.12 de-quantization 117
5.13 reconstructing from noisy measurements 118
5.14 Analysis of optimized measurement vectors 121
5.15 Optimal measurement for perceptual loss 122
5.16 Reconstruction from optimal measurements 123
5.17 Performance of the optimized linear measurements 124

A.1 First order analysis of the residual of different architectures 132
A.2 Effect of net bias on performance for different architectures 133
A.3 Effect of net bias on performance in terms of SSIM for different architectures . . . 133
A.4 Visualization of left singular vectors of the Jacobian for different architectures . . 134

A.5 Analysis of the SVD of the Jacobian of BF-CNN for ten natural images 135

B.1 BF-CNN denoisers trained on $\mathbf{C}^\alpha$ images 138

B.2 Geometric-adaptive harmonic basis for three test images from $\mathbf{C}^\alpha$ class 139

B.3 An additional example of a C^α test image 139

C.1 Additional super-resolution examples. 151

C.2 Additional super-resolution examples. 152

C.3 Additional super-resolution examples. 153

List of Tables

5.1 Spatial super-resolution performance over Set5 and Set14 106

5.2 Spectral super-resolution (deblurring) performance over Set5 106

5.3 Compressive sensing performance over Set68 106

5.4 Run time, in seconds, and (number of iterations until convergence) for different applications 119

List of Appendices

A Learning image densities from data . . . 129

B Generalization beyond memorizing training images . . . 136

C Muli-scale Local conditional image density . . . 148

1 Introduction

Digital images live in a high-dimensional space. We can associate with this space a probability model that assigns high probability to natural images and low probability to distorted images or random pixel configurations. Many problems in image processing and computer vision rely, explicitly or implicitly, on image density models. Hence, image density models are of fundamental importance. But for high-dimensional signals, such as photographic images, density estimation is notoriously difficult due to "the curse of dimensionality": The number of data points needed to estimate a probability distribution from data, in a brute force manner, grows linearly with volume, so exponentially with dimensionality. For example, for 8-bit images of size 64×64, we need enough data to approximate a histogram with 256^{6400} bins. This is far larger than any feasible set of training examples. Certainly way larger than all photographs taken since the invention of camera.

Traditionally, image density estimation is made tractable by factorizing the joint density into a product of low-dimensional parametric models. Although images are high-dimensional, they inhabit a low-dimensional portion of the space thanks to their highly structured nature. Classical models take advantage of such implicit low-dimensionality, which makes natural signals compressible, hence the density factorization possible.

Over a hundred years of steady progress in image density estimation was driven by the interplay between discovering more intricate image statistical structures and engineering better image processing solutions. Through empirical observations, we formed intuitions about properties of image densities, incorporated them into our parametric density models using mathematical formalization, and then tested them by using them to solve various inverse problems, which generally refer to recovering the image from a partial measurement. This process of *scientific experimentation* was the bedrock for developing more sophisticated models: understanding structural properties of images led to improved engineering, which in turn solidified our understanding of the properties of the natural visual world.

1.1 Classical image density models

1.1.1 Gaussian prior

One of the earliest density models assumed over images was Gaussian. An important property of images is their invariance to global translation, i.e. a shifted version of an image is still an image. This implies that the density is stationary and the covariance matrix of the Gaussian prior is convolutional, therefore diagonalizable in the Fourier basis. Additionally, the energy of natural images is highly concentrated in their low-frequency components, due to the prevalence of smooth regions. This observation led to a Gaussian prior in Fourier domain, in which variance falls inversely with spatial frequency. This is commonly referred to as the $1/f$ spectral prior, where f is the spatial frequency.

Although convenient to use, the Gaussian prior results in poor performance in practice. To utilize a prior density in practice, one can obtain a posterior distribution by combining the prior with the likelihood that instantiates some measurement procedure. Then given the measurements, an expected loss is minimized over the posterior to solve an inverse problem. Here, we refer to inverse problems in the broadest sense to include nonlinear and stochastic measurement procedures. Under the Gaussian $1/f$ prior, the optimal estimator for a noisy signal in terms of minimum mean squared error (MMSE) is the Wiener filter (Wiener, 1950). It operates by mapping the noisy image to the frequency domain, shrinking the amplitude of all components, and mapping back to the signal domain. Due to the $1/f$ prior, the high-frequency components are shrunk more aggressively than the lower-frequency components. The outcome of Wiener filtering is an over-smoothed image, with blurred edges and lost fine-scale details. The poor performance of the Wiener filter is an indication that the Gaussian $1/f$ density model is not sufficient to capture the underlying image distribution. More specifically, since it only asserts a prior on the marginal distribution of frequency amplitudes, it falls short of capturing important higher order amplitude dependencies as well as phase dependencies. Since the Fourier basis consists of global sinusoidal functions, this prior does not preserve enough information about the local relationships between pixels in an image.

1.1.2 Sparse prior

In the late twentieth century, an important breakthrough in developing better image density models emerged by the invention of multi-scale wavelet representations (Burt and Adelson, 1983; Mallat, 1989). To alleviate the loss of spatial dependencies, wavelet representations compromise between capturing spatial and frequency information. The finer scale coefficients represent a wide band of high frequencies with higher spatial precision. On the other hand, coarser scale coefficients include a narrow band of low frequencies at the expense of spatial information. This trade-off is depicted in Figure 1.1. Since wavelet coefficients retain some spatial information, their marginal distributions are sparser compared to the Fourier coefficients (Burt and Adelson, 1983; Field, 1987). That is, it is easier to capture the inherent low dimensional nature of images in the wavelet domain. Marginal distributions of wavelet coefficients are typically modeled by a generalized Laplacian distribution $e^{-|x/s|^{\rho}}$, which is a sparse heavy-tailed density (Mallat, 1989).

When used in practice to solve inverse problems, the sparse prior leads to improved performance over the Gaussian prior, which is a direct consequence of the more compactness representation. For example, for the Laplacian prior with $\rho = 1$ the MMSE estimator of a noisy image is

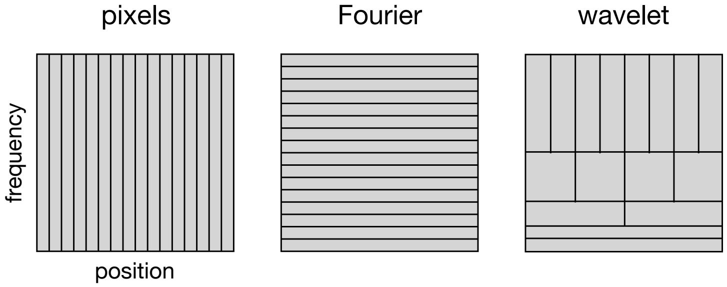


Figure 1.1: Position-frequency partitions of different representations. All rectangular regions have the same area.

achieved by a soft thresholding operator (Mallat, 1989). For the Laplacian prior with $\rho < 1$ the MMSE estimator is a nonlinear smooth function space(Simoncelli and Adelson, 1996) which can be approximated with a hard thresholding function (Donoho, 1995), where wavelet coefficients above a certain threshold are preserved and everything below that threshold is suppressed.

Although a significant improvement over the Gaussian prior, modeling image densities solely based on marginal distributions of the wavelet coefficients is still overly simplistic. Coefficients within and across subbands as well as coefficients across scales are not independent (Buccigrossi and Simoncelli, 1997). On the contrary, there are important higher order dependencies which are entirely lost in marginal wavelet density models. But capturing these important structural properties requires modeling conditional densities which leads to an explosion of model dimensionality. A well-known methodology for capturing conditional dependencies while avoiding the curse of dimensionality is to assume a Markov property.

1.1.3 Markov Priors

Assuming Markov structure is a classic method for capturing global dependencies using chains of low-dimensional local models. Markov random fields (MRF) assume localized conditional dependencies, which guarantee that the density can be factorized into terms acting on local, typically overlapping neighborhoods (Clifford and Hammersley, 1971). Global dependencies arise through the cascaded effects of these local interactions. The idea of using MRF for image pixel modeling goes back to 1970s (Besag, 1974), but it was a paper by Geman and Geman (1984) that popularized the Markov models. Many variants of Markov models were proposed in the subsequent three decades, resulting in steady improvement of image density models.

When MRF models are defined in pixel domain, low dimensional densities over local neighborhoods usually express some smoothness property among neighboring pixels. A common feature shared across all of these models is their hierarchical nature (Bouman and Shapiro, 1994; Freeman and Liu, 2011; Geman and Geman, 1984). Hierarchies facilitate handling larger structures by giving the model the chance to break the smoothness enforced by simple local densities whenever needed, for example around edges. More sophisticated MRF models were later defined on wavelet lattices, in a muti-scale (hierarchical) fashion (e.g., (Buccigrossi and Simoncelli, 1999a;b; Chambolle et al., 1998; Crouse et al., 1998; Şendur and Selesnick, 2002; Cui and Wang, 2005; Lyu and Simoncelli, 2009; Malfait and Roose, 1997; Mihçak et al., 1999; Paget and Longstaff, 1998; Portilla et al., 2003a; Wainwright et al., 2001)). Interestingly, in these models, neighborhoods were reconfigured to include closest coefficients across scale, which resulted in considerable improvement in performance. This change was inspired by the empirical observation that the correlation between the amplitude of a given coefficient and its parent is quite significant (Buccigrossi and Simoncelli, 1999a).

Similarly to marginal models, to test the quality of the density parameterized by an MRF, the model can be used for inference problems, such as restoration, segmentation and synthesis. Since these tasks critically depend on the underlying density of images, a good density model will lead to good performance. Generally, these models have been very effective in modeling texture distributions where stationarity over local neighborhoods is a reasonable assumption (Paget and Longstaff, 1998). Beyond textures, however, the chaining of short-range dependencies has been insufficient to capture the complexity of long-range geometrical structures. An obvious way to address this problem is to increase the size of the neighborhoods over which the local density models are defined. Increasing the size of the neighborhood allows for potentially more complex densities, but it comes with a huge computational cost. As a result, it degrades the capability of MRFs in resolving the curse of dimensionality.

Almost three decades of research on MRFs, which led to a steady and slow progress in probabilistic image modeling, was eclipsed by the advent of deep learning. In chapter 4, we will come back to the fundamental concepts underlying MRF models —specifically stationarity, locality and multi-scale conditioning— and will ask if they are still relevant in the era of deep neural network.

1.2 Image density models in the era of deep learning

1.2.1 A paradigm shift

As highlighted in the brief review of classical density models, the interplay between understanding image properties and engineering advancements was a critical and recurring theme in designing image priors. In the era of deep learning, this paradigm has shifted. Increasingly, we have access to unprecedented computational resources and tools to directly solve individual image processing and computer vision problems, without mention of an image prior.

Generally, the goal is to find a non-linear function, parameterized by a deep neural network, which maps a distorted image to a reconstructed image (in image processing tasks) or a clean image to a characteristic of an image, such as a label (in computer vision tasks). Solving for the parameters is a nonlinear regression problems. One minimizes a loss function over the space of parameters of the deep network usually through backpropagation. Deep net architectures along with stochastic gradient descent optimization procedures have proven to be remarkably powerful for approximating these non-linear functions, as evidenced by their performance across a large variety of tasks. Does this impressive success in learning deep net mappings make the study of image densities irrelevant?

1.2.1.1 Generalization failures

Despite its phenomenal achievements, this approach suffers from a major drawbacks: *generalization failures*. A prior density encapsulates our knowledge about common structures of images and acts as an umbrella that links together the individual tasks. In theory, a good image prior can be used universally to solve *any* inverse problem. On the contrary, in a regression-based approach, each task is viewed as a separate problem to be solved by finding a new set of optimal parameters, which is computationally costly. So, the impressive empirical performance comes at the price of specializing and overfitting the model to a particular task. Research areas like transfer learning (Weiss et al., 2016) and meta learning (Lemke et al., 2015) emerged to address the issue of generalization failure across tasks.

In addition to the inter-task generalization failures, deep net models trained to solve the same problem are prone to overfit to the particular settings of the problem. This issue arises from hyper-parameter tuning due to the high expressivity of deep neural networks. Techniques like ensemble methods (Ganaie et al., 2022) and domain adaptations (Farahani et al., 2021) aim to resolve overfitting to datasets and hyper-parameters within a given task.

1.2.1.2 Shallow understanding of deep models

Failure of generalization is tightly connected to our *lack of scientific understanding* of why and how deep neural nets work. In the classical approach, we incorporated our hypotheses into priors, which were then used to solve engineering problems. As a result, performance improvements were directly linked to our understanding of the visual world, and served to confirm or falsify our hypotheses in the form of prior models. Today, our engineering advancements do not necessarily translate to deeper understanding of our models. Superior performance is usually achieved through a huge number of trial-and-error iterations, resulting in complicated oversized models with many pieces whose contributions are not immediately obvious.

This is not only intellectually unsatisfying, but also leaves us blind to potential flaws of our models. For example, Szegedy et al (Szegedy et al., 2013) showed that deep net classifiers, despite their excellent performance on the test set, are not robust to imperceptible perturbations of input images (known as adversarial examples). In another study, Geirhos et al (Geirhos et al., 2018) showed that these classification networks use features inessential to the identity of the objects, such as texture as opposed to shape. Along the same lines (Beery et al., 2018; Geirhos et al., 2020; Rosenfeld et al., 2018; Zech et al., 2018), it has been shown that object detector networks can miss relevant objects in the scene due to their reliance on spurious correlations. For example, the network fails to identify a camel on a green background because it relies on the yellow sandy background usually accompanying a camel in the training set examples. So, as these failure cases highlight, not only do we fail to generalize outside of the training regime (slightly perturbed images, shapes with unusual textures, or objects on unlikely backgrounds), but also we fail to predict when these failures occur. Unlike the classical approach, the superior performance of deep nets does not come with guarantees for success and it is not clear which pieces are essential to build on. Techniques like ablation studies (Meyes et al., 2019) and sub-fields like mechanistic interpretability (Bereska and Gavves, 2024) emerged to address these shortcomings, but so far our understanding of deep models remains shallow.

1.2.2 Extracting an image prior from a deep net denoiser

Robust generalization and interpretability are valuable implications of using image priors in the classical framework. The existing prior models, however, have proven to be subpar compared to the state-of-the-art deep net models in virtually every sub-field of computer vision and image processing. So, is it possible to revisit the study and use of image densities and leverage the unprecedented power of deep learning for image densities estimation? The short answer is yes, and it involves reversing the order of steps in the classical framework. Rather than explicitly defining a parametric image prior to solve inverse problems, we solve an inverse problem by learning a deep network model from data and then we extract the prior implicit in this learned network.

For this purpose, we choose the task of denoising. In recent years, deep neural network denoisers have exhibited phenomenal performance (Ilesanmi and Ilesanmi, 2021; Zhang et al., 2017a). Although these denoisers do not explicitly rely on an image prior, their incredible performance shows they can differentiate between image and noise, implying they have some "knowledge" of what is structure and what is randomness. In other words, they implicitly embed an image prior. Gaussian denoising is particularly suitable for our purpose because the connection between denoising mapping and implicit prior is clear due to a classical statistics result, attributed to Tweedie and Miyasawa (Miyasawa, 1961; Robbins, 1956a). It shows that a denoiser optimized to minimize squared error is in fact computing the gradient of the log of the density of noisy images (also known as the score). This result provides a direct relationship between least-squares optimal denoising and the embedded prior (Raphan and Simoncelli, 2011), which allows the extraction of a prior from a deep net denoiser. We developed (Kadkhodaie and Simoncelli, 2021a; 2020) an Iterative Score Ascent (ISA) algorithm by using this denoiser-estimated gradient to draw high-probability samples from the embedded prior. In Chapter 2, we discuss our sampling method extensively. Additionally, we flesh out similarities and differences between our method and two closely related generative models, namely score-based generative model (Song and Ermon, 2019a) which inspired our work, and diffusion models (Ho et al., 2020) which have become the most popular score-based sampling method.

1.2.2.1 What is the implicit prior?

The last few years have witnessed an explosion of research on score-based image generative models, under the umbrella term of "diffusion models". The quality of samples generated by these models is usually superior to any classical density model, which lends them a lot of traction. But, unlike the classical framework, we only observe individual images sampled from the prior, without having direct access to the density. As a result, the nature of these powerful priors remains a mystery. This gives rise to a crucial question: *what is the prior?* The rest of this thesis explores this question. In our studies, we use the *scientific method* to understand, improve and utilize the implicit prior. We form hypotheses based on empirical observations and then design controlled experiments to verify or falsify them. In all of these experiments, we look for improvements in *generalization*, in some sense of the word, as our measure of success.

We start by proposing an architectural modification in the network in order to make the denoiser algebraically homogeneous. This idea is based on a simple intuition that the embedded image prior should be invariant to global intensity changes. The intensity invariance property is implemented by removing all the additive constant terms from the architecture which renders the model locally linear. As a result, the Deep Neural Network (DNN) denoiser generalizes far beyond the training noise variance range, which has important implications for the sampling algorithm. Additionally, locally linear networks can be analysed by linear algebraic tools. In the second half of Chapter 2 we present these results and show that the denoising operation can be interpreted as an approximate projection onto an adaptive image subspace whose dimensionality falls with noise variance.

Next, we study the continuity of the implicit image prior. Simple thought experiments suggest that images are concentrated on or near low-dimensional continuous manifolds whose local coordinates represent deformations and intensity variations. For a given photograph, applying local continuous deformations (e.g., translations, rotations, dilations, intensity changes) yields a low-dimensional family of natural-appearing images, that trace out a low-dimensional, highly curved, manifold in the space of pixels. Does the learning procedure succeed in approximating a continuous prior by interpolating between the training examples? Or is it doomed by the curse of dimensionality to only arrive at a discrete set of delta functions corresponding to a memorized set of training examples? To answer this question, we design an experiment to study the behavior of the model as a function of the training set size. We find that the learned prior is in fact a discrete set of delta functions when the training set is small, but it quickly enters a transition phase as the training set size grows. With large enough datasets, the network generalizes by interpolating between the training examples, evidenced by high quality novel generated samples. Surprisingly, we observe that, when the number of training images is large enough, two models trained on non-overlapping subsets of a dataset learn nearly the same density.

This strong generalization result, combined with the high visual quality of the sampled images, suggests that the learned prior is a good approximation of the underlying density. But how do these models achieve this approximation despite the "curse of dimensionality"? Answering this question requires understanding the alignment between the "true" density and the inductive biases of the network. Using linear algebraic tools mentioned before, we analyze the learned denoising functions and show that, consistent with the classical description of denoising, the inductive biases give rise to a shrinkage operation in a basis adapted to the underlying image. Examination of these bases reveals oscillating harmonic structures along contours and in homogeneous regions. We demonstrate that trained denoisers are inductively biased towards these geometry-adaptive harmonic bases since they arise not only when the network is trained on photographic images, but also when it is trained on image classes supported on low-dimensional manifolds for which the harmonic basis is suboptimal. Chapter 3 is dedicated to describing these results.

Having established that a DNN denoiser can generalize beyond memorizing the training set, we now use this framework to return to and study the question of low-dimensionality of image densities. A critical feature for the success of DNNs in the diffusion framework is their global receptive fields (RF). Large RF allows for capturing long range dependencies in the image through modeling the joint density directly. Naturally, this yields to a large number of parameters, currently on the order of hundreds of millions for the state-of-the-art diffusion models. Can we exploit image properties to factorize the joint density into low dimensional densities, thereby reducing the number of parameters and training examples?

Before the emergence of deep learning, random field models aimed at capturing the low dimensional structure of image densities by assuming the Markov property. But, MRFs have not succeeded in modeling image densities beyond textures. In Chapter 4, we leverage the expressivity of DNN to build a powerful image density model assuming a multi-scale Markov property. We develop a low-dimensional probability model for images decomposed into multi-scale wavelet sub-bands. The image density 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 density can then be estimated with a conditional CNN (cCNN) with a small receptive field (RF). But unlike classical MRFs, we are not computationally constrained to 3×3 Markov neighborhoods or to simple parametric forms for the local interaction potentials, which allows for a more sophisticated density.

An important feature of this model is its "foveated" nature. The effective size of the neighborhood grows with scale, from small in fine scales to large in coarser scales, reaching a global RF at the coarsest scale, 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. We then evaluate this low-dimensional density model by using it for denoising, super-resolution, and synthesis. We show that locality and stationarity assumptions hold for conditional RF sizes as small as 9×9 without harming performance. This can be viewed as a powerful alternative path for improving score-based density estimation instead of increasing the size of the model and training set. MRFs in conjunction with the rich expressivity of DNNs alleviate the curse of dimensionality by learning low-dimensional approximations of high-dimensional image probabilities.

Finally, in Chapter 5, we put the denoiser prior into use in solving linear inverse problems. We describe a generalization of our sampling algorithm to constrained sampling, which provides a method for using the implicit prior to solve *any* linear inverse problem, with no additional training, thus extending the power of supervised learning for denoising to a much broader set of problems. The algorithm relies on minimal assumptions and exhibits robust convergence over a wide range of parameter choices. To demonstrate the generality of our method, we use it to obtain state-of-the-art levels of unsupervised performance for inpainting, deblurring, super-resolution, compressive sensing, and colorization. The method described in Chapter 5 was the first algorithm to use the prior embedded in a denoiser for solving inverse problems in a stochastic fashion.

Deep learning has equipped us with tools and techniques for solving problems which used to be deemed infeasible or impossible. Learning image densities is one of those problems. This thesis is focused on methods to not only learn but also examine, interpret and utilize these powerful image priors. In the discussion section, I describe some ongoing work and a few future directions among numerous possibilities that have emerged in the era of deep learning.

2 Learning image densities from data using a denoiser

2.1 Image prior implicit in a denoiser

Nearly all problems in image processing and computer vision have been revolutionalized in recent years by the use of deep Convolutional Neural Networks (CNNs). These networks are generally optimized in supervised fashion to obtain a direct input-output mapping for a specific task. This approach does not explicitly rely on a known prior, and offers performance far superior to classical prior-based methods. The superior performance of CNNs suggests that they embed, implicitly, sophisticated prior knowledge of images. These implicit priors arise from a combination of the distribution of the training data, the architecture of the network (Ulyanov et al., 2020), regularization terms included in the optimization objective, and the optimization algorithm.

Here, our goal is to extract the implicit prior from a network trained for denoising. We choose denoising not because it is of particular importance or interest, but because we can make the relationship between mapping of a denoiser and a prior explicit. We start with a result

Section 2.1 to section 2.2 were presented in an arXiv paper (Kadkhodaie and Simoncelli, 2020), and later published in (Kadkhodaie and Simoncelli, 2021a). An implementation of the algorithm for sampling and code for generating results are available at https://github.com/LabForComputationalVision/universal_inverse_problem.

from classical statistics (Miyasawa, 1961; Robbins, 1956b) that states that a denoiser that aims to minimize squared error of images corrupted by additive Gaussian noise may be interpreted as computing the gradient of the log of the density of noisy images. This result is related to Score Matching (Hyvärinen and Dayan, 2005), but provides a more direct relationship between least-squares optimal denoising and the embedded prior (Raphan and Simoncelli, 2011; Reehorst and Schniter, 2019). We develop a Stochastic Iterative Score Ascent algorithm (SISA) that uses this denoiser-estimated gradient of log probability (aka score) to draw high-probability samples from the embedded prior. Importantly, we use a *blind* denoiser that can handle noise contamination of unknown amplitude, which provides a means of adaptively controlling the gradient step sizes and the amplitude of injected noise, enabling robust and efficient convergence.

2.1.1 Image priors, manifolds, and noisy observations

Digital photographic images lie in a high-dimensional space ($\mathbb{R}^N$, where N is the number of pixels), and simple thought experiments suggest that they are concentrated on or near low-dimensional manifolds whose local coordinates represent continuous deformations and intensity variations. In contrast, images generated with random pixels are almost always feature and content free, and thus not considered to be part of this manifold. We can associate with this manifold a prior probability model, $p(x)$, by assuming that images within the manifold have constant or slowly-varying probability, while unnatural or distorted images (which lie off the manifold) have low or zero probability. Suppose we make a noisy observation of an image, $y = x+z$, where $x \in R^N$ is the original image drawn from $p(x)$, and $z \sim \mathcal{N}(0, \sigma^2 I_N)$ is a sample of Gaussian white noise. The observation density $p(y)$ is related to the prior $p(x)$ via marginalization:

$$p(y) = \int p(y|x)p(x)dx = \int g(y-x)p(x)dx, \tag{2.1}$$

where $g(z)$ is the Gaussian noise distribution. Equation (2.1) is in the form of a convolution, and thus $p(y)$ is a Gaussian-blurred version of the signal prior, $p(x)$. Moreover, the family of observation densities over different noise variances, $p_\sigma(y)$, forms a Gaussian scale-space representation of the prior (Koenderink, 1984; Lindeberg, 1994), analogous to the temporal evolution of a diffusion process.

2.1.2 Least squares denoising and CNNs

Given a noisy observation, y, the minimum mean squared error (MMSE) estimate of the true signal is well known to be the conditional mean of the posterior density:

$$\hat{x}(y) = \int x p(x|y) dx = \int x \frac{p(y|x)p(x)}{p(y)} dx \tag{2.2}$$

The structure of the equation mirrors the traditional approach to the problem: one chooses a prior probability model, $p(x)$, combines it with a likelihood function describing the noisy measurement process, $p(y|x)$, and solves. Modern denoising solutions, on the other hand, are often based on supervised learning of a direct mapping from noisy to denoised images. One expresses the estimation function (as opposed to the prior) in parametric form, and sets the parameters by minimizing the denoising MSE over a large training set of example signals and their noise-corrupted counterparts (Burger et al., 2012; Elad and Aharon, 2006; Hel-Or and Shaked, 2008; Jain and Seung, 2009). Current state-of-the-art denoising results using CNNs obtained with this supervised approach are far superior to results of previous methods (Chen and Pock, 2017; Huang et al., 2017; Zhang et al., 2017a). These architectures can be simplified by removing all additive bias terms, with no loss of performance (Mohan* et al., 2020). The resulting *bias-free* networks offer two important advantages. First, they automatically generalize to all noise levels, even when trained on a narrow range of noise. A network trained on images with barely noticeable levels of noise can produce high quality results when applied to images corrupted by noise of any amplitude. Second, they may be analyzed as adaptive linear systems, which reveals that they perform an approximate projection onto a low-dimensional subspace. In our context, we interpret this subspace as a tangent hyperplane of the image manifold. Moreover, the dimensionality of these subspaces falls inversely with σ, and for a given noise sample, the subspaces associated with different noise amplitude are nested, with high-noise subspaces lying within their lower-noise counterparts. In the limit as the noise variance goes to zero, the subspace dimensionality grows to match that of the manifold at that particular point. We take a more in-depth look at the properties and analysis of these denoisers in section 2.3.

2.1.3 Exposing the implicit prior through Empirical Bayes estimation

Trained CNN denoisers contain detailed prior knowledge of image structure, but Eq. (2.2) suggests that it is embedded within a high-dimensional integral. How can we make use of this implicit prior? Recent results have derived relationships between Score Matching density estimates and denoising, and have used these relationships to make use of implicit prior information (Bengio et al., 2013; Li et al., 2019; Saremi and Hyvarinen, 2019; Song and Ermon, 2019a). Here, we exploit a more direct but less-known result from the literature on Empirical Bayesian estimation. The idea was introduced in (Robbins, 1956a), extended to the case of Gaussian additive noise in (Miyasawa, 1961), and generalized to many other measurement models (Raphan and Simoncelli, 2011). In the case of additive Gaussian noise, one can rewrite the estimator of Eq. (2.2) as:

$$\hat{x}(y) = y + \sigma^2 \nabla_y \log p(y). \tag{2.3}$$

The proof is relatively straightforward. The gradient of the observation density of Eq. (2.1) is:

$$\nabla_y\, p(y) = \frac{1}{\sigma^2} \int (x-y) g(y-x) p(x) dx = \frac{1}{\sigma^2} \int (x-y) p(y,x) dx.$$

Multiplying both sides by $\sigma^2/p(y)$ and separating the right side into two terms gives:

$$\sigma^2 \frac{\nabla_y \, p(y)}{p(y)} = \int xp(x|y)dx - \int yp(x|y)dx = \hat{x}(y) - y.$$

Rearranging terms and using the chain rule to compute the gradient of the log gives Eq. (2.3). This remarkable result re-expresses the integral over the prior and likelihood of Eq. (2.2) in terms of a *gradient*. Note that 1) the relevant density is not the prior, $p(x)$, but the noisy *observation density*, $p(y)$; 2) the gradient is computed on the *log* density (the associated "energy function"); and 3) the gradient adjustment is *not* iterative - the estimate is achieved in a single step, and holds for any noise level, σ.

2.2 Drawing high-probability samples from the implicit prior

Suppose we wish to draw a sample from the prior implicit in a denoiser. Equation (2.3) allows us to generate an image proportional to the gradient of $\log p(y)$ by computing the denoiser residual, $f(y) = \hat{x}(y) - y$. Song and Ermon (Song and Ermon, 2019a) developed a Markov chain Monte Carlo (MCMC) scheme, combining gradient steps derived from Score Matching and injected noise in a Langevin sampling algorithm to draw samples from a sequence of densities $p_\sigma(y)$, while reducing σ in a sequence of discrete steps, each associated with an appropriately trained denoiser. In contrast, starting from a random initialization, y_0, we aim to find a *high-probability* image (i.e., an image from the manifold) using a more direct and efficient stochastic gradient ascent procedure.

2.2.1 Unconstrained sampling algorithm

We compute gradients using the residual of a universal blind CNN denoiser, which automatically estimates and adapts to each noise level. On each iteration, the algorithm takes a small step in the direction specified by the denoiser, moving toward the image manifold and thereby reducing the amplitude of the effective noise. Under the interpretation that the denoiser performs a projection onto the current approximation of the image manifold, this noise reduction occurs in the subspace orthogonal to that manifold, and noise components parallel to the manifold are retained. As the effective noise decreases, the observable dimensionality of the image manifold increases (Mohan* et al., 2020), enabling the synthesis of detailed image content. Since the family of observation densities, $p_\sigma(y)$ forms a scale-space representation of $p(x)$, the algorithm may be viewed as a form of coarse-to-fine optimization (Blake and Zisserman, 1987; Geman and Geman, 1984; Kirkpatrick et al., 1983; Lucas and Kanade, 1981). Assuming the step sizes are adequately controlled, the procedure will converge to a local optimum of the implicit prior - i.e., a point on the manifold. Figure 2.1 provides a visualization of this process in two dimensions.

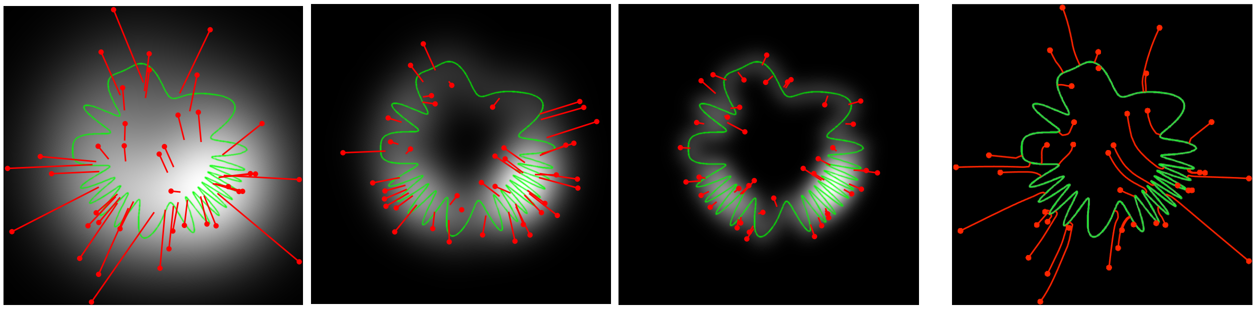


Figure 2.1: Two-dimensional simulation/visualization of the sampler. Forty example signals x are sampled from a uniform prior on a manifold (green curve). First three panels show, for three different levels of noise, the noise-corrupted measurements of the signals (red points), the associated noisy signal distribution $p(y)$ (indicated with underlying grayscale intensities), and the least-squares optimal denoising solution $\hat{x}(y)$ for each (end of red line segments), as defined by Eq. (2.2), or equivalently, Eq. (2.3). Right panel shows trajectory of our SISA algorithm (Algorithm 1), starting from the same initial values y (red points) of the first panel. Algorithm parameters were $h_0 = 0.05$ and $\beta = 1$ (i.e., no injected noise). Note that, unlike the single-step least-squares solutions, the iterative trajectories are curved, and always arrive at solutions on the signal manifold.

Each iteration operates by taking a deterministic step in the direction of the gradient (as obtained from the denoising function) and injecting some additional noise:

$$y_t = y_{t-1} + h_t f(y_{t-1}) + \gamma_t z_t, \tag{2.4}$$

where $f(y) = \hat{x}(y) - y$ is the residual of the denoising function, which is proportional to the gradient of $\log p(y)$, from Eq. (2.3). The parameter $h_t \in [0, 1]$ controls the fraction of the denoising correction that is taken, and γ_t controls the amplitude of a sample of white Gaussian noise, $z_t \sim \mathcal{N}(0, I)$. The purpose of injecting noise is two-fold. First, from an optimization perspective, it allows the method to avoid getting stuck in local maxima. Second, it allows stochastic exploration of the manifold, yielding a more diverse (higher entropy) family of solutions. The effective noise variance of image y_t is:

$$\sigma_t^2 = (1 - h_t)^2 \sigma_{t-1}^2 + \gamma_t^2, \tag{2.5}$$

where the first term is the variance of the noise remaining after the denoiser correction (assuming the denoiser is perfect), and the second term is the variance arising from the injected noise. The assumption of perfect denoising is an idealization, but we show empirically (Fig. 2.2) that SISA algorithm converges reliably, with error levels falling as predicted by Eq. (2.5) or faster, across different settings of β and h_0.

To ensure convergence, we require the effective noise variance on each time step to be reduced, despite the injection of additional noise. For this purpose, we introduce a parameter $\beta \in [0, 1]$ to control the proportion of injected noise ($\beta = 1$ indicates no noise), and enforce the convergence by requiring that:

$$\sigma_t^2 = (1 - \beta h_t)^2 \sigma_{t-1}^2. \tag{2.6}$$

Combining this with Eq. (2.5) yields an expression for γ_t in terms of h_t:

$$\gamma_t^2 = \left[(1 - \beta h_t)^2 - (1 - h_t)^2\right] \sigma_{t-1}^2 = \left[(1 - \beta h_t)^2 - (1 - h_t)^2\right] \left\| f(y_{t-1}) \right\|^2 / N,$$

where the second equation assumes that the magnitude of the denoising residual provides a good estimate of the effective noise standard deviation, as was found in (Mohan* et al., 2020). This allows the denoiser to adaptively control the score ascent step sizes, reducing them as the y_t approaches the manifold (see Fig. 2.1). This automatic adjustment results in efficient and reliable convergence, as demonstrated empirically in Fig. 2.2. Our initial implementation with a small constant fractional step size $h_t = h_0$ produced high quality results, but required many iterations. Intuitively, step sizes that are a fixed proportion of the distance to the manifold lead to exponential decay - a form of Zeno's paradox. To accelerate convergence, we introduced a schedule for increasing the step size proportion, starting from $h_0 \in [0, 1]$. The sampling process is summarized in Algorithm 1.

Algorithm 1: Stochastic Iterative Score Ascent (SISA) method for sampling from the implicit prior of a denoiser, using denoiser residual $f(y) = \hat{x}(y) - y$.

```
parameters: $\sigma_0$, $\sigma_L$, $h_0$, $\beta$
initialization: $t = 1$, draw $y_0 \sim \mathcal{N}(0.5, \sigma_0^2 I)$
while $\sigma_{t-1} \geq \sigma_L$ do
    $h_t = \frac{h_0 t}{1+h_0(t-1)}$;
    $d_t = f(y_{t-1})$;
    $\sigma_t^2 = \frac{||d_t||^2}{N}$;
    $\gamma_t^2 = \left((1 - \beta h_t)^2 - (1 - h_t)^2\right) \sigma_t^2$;
    Draw $z_t \sim \mathcal{N}(0, I)$;
    $y_t \leftarrow y_{t-1} + h_t d_t + \gamma_t z_t$;
    $t \leftarrow t + 1$
end
```

2.2.2 Image synthesis examples

For the denoiser, we used BF-CNN (Mohan* et al., 2020), a bias-free variant of DnCNN (Zhang et al., 2017a). We obtained similar results (not shown) using other CNN architectures described in (Mohan* et al., 2020), including Recurrent-CNN, Dense-Net, and truncated U-Net. We trained this network on three different datasets: 40 × 40 patches cropped from Berkeley segmentation training set (Martin et al., 2001), in color and grayscale, and MNIST dataset (LeCun and Cortes, 2010) (see section 2.3 for further details). We chose parameters $\sigma_0 = 1$, $\sigma_L = 0.01$, and $h_0 = 0.01$. Figures 2.3, 2.4, and 2.5 provide visualization of example trajectories. Additional visual examples, obtained with different levels of β, are shown in Figure 2.6 and Figure 2.7.

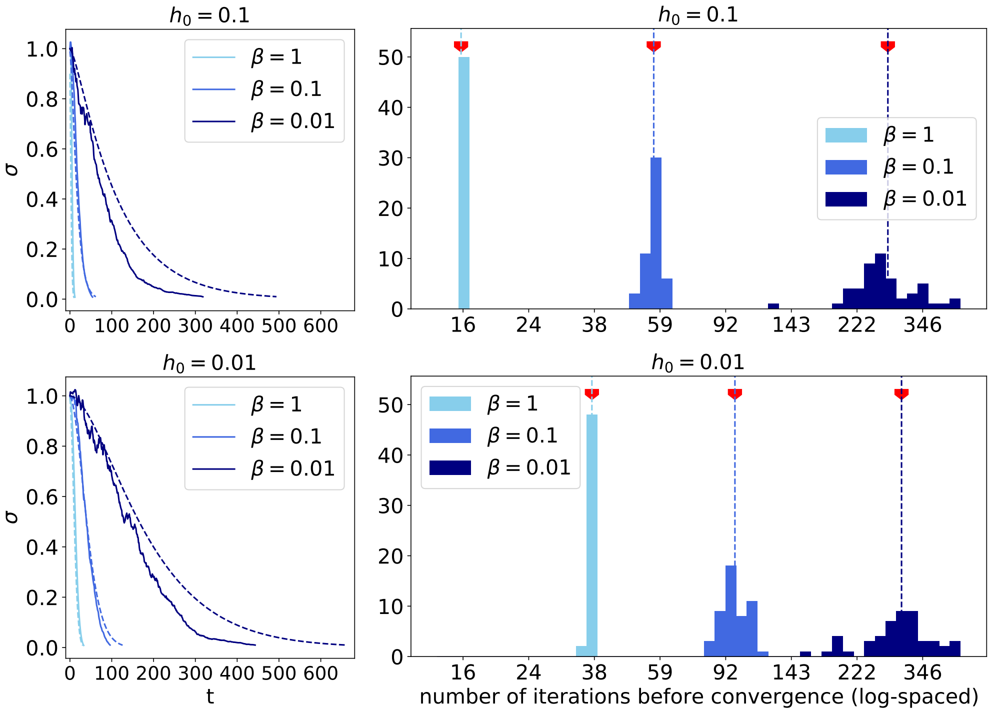


Figure 2.2: Convergence of SISA sampling algorithm, quantified in terms of the effective noise standard deviation $\sigma = \frac{||d_t||}{\sqrt{N}}$ for three different values of β and two values of h_0. **Left.** Convergence of single examples (solid curves) is well-behaved and efficient in all cases. Dashed curves indicate the convergence predicted from the formulation of SISA algorithm: $\sigma_t = (1 - \beta h_t)\sigma_{t-1}$. For $\beta = 1$ (no injected noise), the empirical convergence closely approximates the prediction. For larger amounts of injected noise (smaller β), convergence is naturally slower, but faster than predicted. This is because, for small noise levels, the magnitude of the denoising residual is an underestimate of the input noise level, which means that γ is the upper bound on the injected noise. Conceptually, this effect is related to the dimensionality of the image subspace tangent to the manifold. The injected noise has non-zero amplitude in all directions, including directions parallel to the tangent subspace. These directions do not contribute to the effective noise variance because they contain image content. The number of these directions increases with reduction of noise (Mohan* et al., 2020) resulting in increasing overestimation of noise variance in the predicted curves. **Right.** Distribution of number of iterations before convergence to a criterion level of $\sigma = 0.01$ for 50 images. Red symbols indicate average values for each β and h_0.

2.2.3 Connection and comparison to related work

2.2.3.1 Denoising score matching

Our method is closely related to recent work that uses Score Matching (Hyvärinen and Dayan, 2005) to draw samples from an implicit prior. This line of work is rooted in the connection between

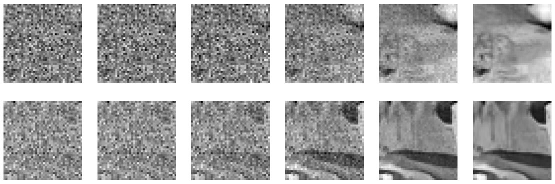


Figure 2.3: Visualization of sampling SISA algorithm trajectories. Each row shows a sequence of images, $y_t, t = 1, 9, 17, 25, \ldots$, from the iterative sampling procedure, with different initializations, y_0, and no added noise ($\beta = 1$), demonstrating the way that SISA algorithm amplifies and "hallucinates" structure found in the initial (noise) images. Convergence is typically achieved in less than 40 iterations with stochasticity disabled.

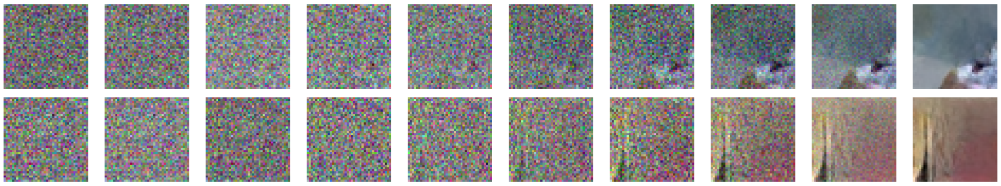


Figure 2.4: Visualization of SISA sampling algorithm trajectories for color images. Sampling from the implicit prior embedded in a BF-CNN denoiser trained on color Berkeley segmentation dataset.

Denoising Autoencoders and Score Matching, first described in (Vincent, 2011a). Denoising autoencoders learn the gradient of log density, and their connection to an underlying data manifold has been explored in (Alain and Bengio, 2014; Bengio et al., 2013; Vincent et al., 2008). More recently, Refs. (Saremi and Hyvarinen, 2019; Saremi et al., 2018) trained a neural network to directly estimate energy (the negative log prior) and interpreted its gradient as the Score of the data. Finally, in a breakthrough paper that partially inspired our approach, Song and Ermon (Song and Ermon, 2019a) trained a sequence of denoisers with decreasing levels of noise and used Langevin dynamics to sample from the underlying probability distributions in succession. Our work differs from these in several important ways:

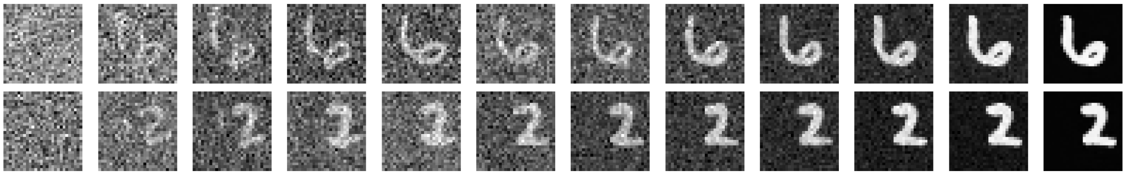


Figure 2.5: Visualization of SISA sampling algorithm trajectories for MNIST. Sampling from the implicit prior embedded in a BF-CNN denoiser trained on MNIST dataset.

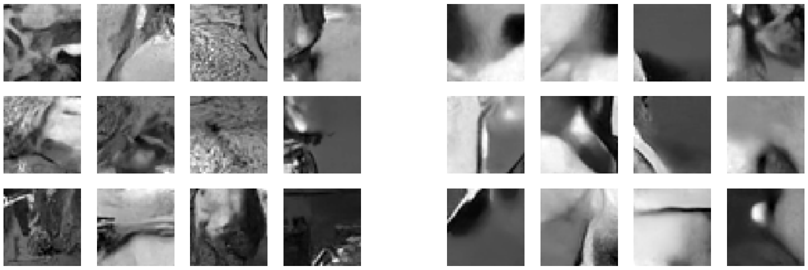


Figure 2.6: The effect of hyper-parameter β on generated samples of natural patches. **Left.** Samples drawn with different initializations y_0, using a moderate level of injected noise ($\beta = 0.5$). Images contain natural-looking features, with sharp contours, junctions, shading, and in some cases, detailed texture regions. **Right.** Samples drawn with more substantial injected noise ($\beta = 0.1$). The additional noise helps to avoid local maxima, and arrives at images that are smoother and higher probability, but still containing sharp boundaries.

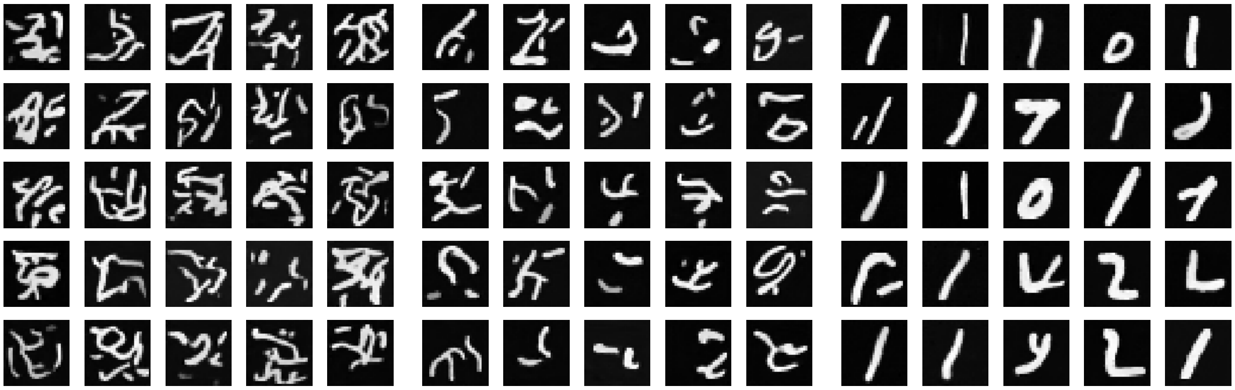


Figure 2.7: The effect of hyper-parameter β on generated samples on MNIST digits. Training BF-CNN on the MNIST dataset of handwritten digits (LeCun et al., 2010) results in a different implicit prior (compare to Figure 2.6). Each panel shows 16 samples drawn from the implicit prior, with different levels of injected noise (increasing from left to right, $\beta \in \{1.0,\ 0.3,\ 0.01\}$).

(1) *Direct derivation.* Our method is based on Miyasawa's explicit relationship between the denoiser mapping and implicit density (Miyasawa, 1961), which can be proven with a few lines of math (Section 2.1.3). The relationship of this result to Score Matching (Raphan and Simoncelli, 2011; Reehorst and Schniter, 2019), and the relationship between Score Matching and Denoising Autoencoders (Vincent, 2011a) are both significantly more nuanced and complex.

(2) *Simplicity and parsimony.* Our method (SISA) describes a coarse-to-fine gradient ascent algorithm, which aims to converge to a local maximum of $p(x)$ in the deterministic case ($\beta = 1$)

and a high probability point on $p(x)$ in the stochastic case ($\beta < 1$), similarly to the general gradient-based framework of optimization in deep learning. The gradient provided by the denoiser is not used for sampling, but for removing a fraction of the noise at each step. Our formulation relies on a single universal blind denoiser which implicitly embeds an infinite family of distributions of noisy images, corresponding to a continuous range of noise levels. In contrast, the algorithm presented in (Song and Ermon, 2019a) uses a discrete sequence of denoisers (precisely 10 denoisers), each trained for a single noise level, and embedding a distribution of noisy images with that specific noise level. Langevin dynamics are used to sample from each of these distributions in succession, with each stage initialized from the sample generated by the previous stage. This procedure is repeated until the noise is sufficiently small (annealed Langevin dynamics). Langevin sampling guarantees (asymptotically) convergence to each successive distribution, although convergence to intermediate distributions is not necessary and the samples drawn from them are redundant.

(3) *Efficiency through step-size adaptivity.* Our SISA algorithm is based on a universal, blind denoiser which estimates the noise level automatically. After each step, the denoiser adapts to the updated implicit noisy density, both in terms of gradient direction and magnitude. Hence the step size is automatically adjusted based on the denoiser's estimate of distance to the manifold. As a result, the number of steps required for convergence is not constant, but is adjusted to the distribution complexity and the specific image under synthesis, as shown in Figure 2.2. In contrast, the algorithm presented in (Song and Ermon, 2019a) assumes a discrete sequence of denoisers trained for a pre-specified sequence of noise levels. This method requires choices of the schedule of noise levels (standard deviations, and number of iterations used at each level). The continuous maximization of probability combined with automatic step size adjustment result in substantial gains in efficiency. For comparison, Song&Ermon (Song and Ermon, 2019a) report 1000 iteration to synthesize a 32×32 image, whereas we synthesize 40×40 images in roughly 35 iteration (for $\beta = 1$).

(4) *Flexible stochasticity of synthesis.* We use the residual magnitude as an estimate of the

implicit noise level as described in equation 2.6, and this is used to control the amplitude of injected noise. So SISA algorithm can go from entirely deterministic beyond the initialization ($\beta = 1$) to increasing levels of stochastic with smaller β. Note that this differs markedly from the control of noise injection in the Langevin method, which is of constant magnitude in each stage.

This is similar to score-based diffusion algorithms (Ho et al., 2020; Song and Ermon, 2019b; 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.

2.2.3.2 Denoising diffusion probabilistic models

Denoising diffusion probabilistic models (DDPM) (Ho et al., 2020) were introduced shortly after we developed SISA algorithm. DDPM has become the most popular and well-known denoising or score based generative model. DDPM gets rid of sequencing, so SISA algorithm is similar to DDPM in using a denoiser trained on a wide range of noise and employing it iteratively to draw samples from the density. Our work is different from this model in the following ways:

(1) *Convergence to true vs. learned density.* DDPM is based on a variational inference description first proposed in (Sohl-Dickstein et al., 2015). A continuous diffusion process is applied to a target density of clean images, assuming it converges to a Gaussian distribution in infinite time. Then a reverse process is applied to the Gaussian distribution to take it to the initial target distribution. The assumption is that for an infinitesimal change in time, which is tied to change in noise variance, the conditional or transition distributions are the same. This is a sufficient (but not necessary) detailed balance condition that guarantees convergence.

In practice however, one must assume a discrete noise schedule that introduces an error to convergence. For image densities in particular, any discretization causes error, because the forward and backward conditional are very different for image densities. The conditional density in the forward pass describes the distribution of a noisy image given a less noisy image which is simply a Gaussian. However, distribution of less noisy images given more noisy image in the reverse process is very non-Gaussian and complex. This conditional can be approximated with a Gaussian only in the limit of very small noise steps. On the contrary, SISA algorithm is proposed as a discrete gradient ascent algorithm which converges to the learned density embedded in the denoiser (as opposed to a true density). We believe this description is not only simpler and more direct, but also more consistent with practical constraints. The true image density is not known and it is far too complex and high dimensional to be described explicitly, hence it is not clear how meaningful the convergence bound guarantees are. On the other hand, the density we have access to and converge to is the learned density embedded in the denoiser. This description leads us to investigate a parallel set of questions involving understanding the properties of the learned density.

(2) *Noise variance estimation error* DDMP models take the noise variance as an input. This value is used to set the step size throughout the synthesis trajectory. This is another source of error since it is not guaranteed that the noise variance is reduced to the expected level to achieve the intended noise variance in the next step. In contrast, SISA algorithm relies on a blind denoiser which estimates the noise level automatically and adjust the step size to the estimated noise level (as opposed to the true noise level remained on the image).

(3) *Initialization error* Another source of error in DDPM is initialization error. Since the network is not blind and universal, in order to approximate the score correctly, it should be trained on all noise levels. This is not possible, so the network does not embed the score for all noise levels. This is particularly problematic for the score at initialization. On the other hand, SISA algorithm is based on a blind and universal denoiser which extrapolates to large noise levels without training (see next section).

2.3 Deep Neural Network Denoisers

At the core of SISA sampling algorithm is a deep neural network denoiser, so we dedicate this section to studying DNN denoisers.

In the past decade, purely data-driven models based on convolutional neural networks (LeCun et al., 2015) have come to dominate all previous methods in terms of performance. These models consist of cascades of convolutional filters, rectifying nonlinearities, skip connections and sometimes down-sampling which are capable of representing a diverse and powerful set of functions. Training such architectures to minimize mean square error over large databases of noisy natural-images achieves current state-of-the-art results (Chen and Pock, 2017; Herbreteau et al., 2024; Huang et al., 2017; Ronneberger et al., 2015a; Zhang et al., 2017a; 2018a).

An important advantage of deep-learning techniques over traditional methodology is that a single neural network can be trained to perform denoising at a wide range of noise levels. This is often achieved by simulating the whole range of noise levels during training (Zhang et al., 2017a). Here, we show that this is not necessary. Neural networks can be made to *generalize automatically across noise levels* through a simple modification in the architecture: removing all additive constants. We find this holds for a variety of network architectures proposed in previous literature. We provide extensive empirical evidence that denoising architectures systematically overfit to the noise levels in the training set, and that this is due to the presence of a net bias. Suppressing this bias makes it possible to attain state-of-the-art performance while training over a very limited range of noise levels. This result has significant implications regarding the initialization assumption in SISA sampling algorithm discussed earlier.

Geometrically, locally linear mapping is motivated by the invariance of the image prior to

Section 2.3 is published (Mohan* et al., 2020) in collaboration with Carlos Fernandez-Granda and Sreyas Mohan. The code for generating results is available at https://github.com/LabForComputationalVision/bias_free_denoising.

intensity changes. For a natural image on the image manifold, its different variants with different intensity are high probability images lying on the manifold. This implies that the image manifold consists of linear facets going through the origin, giving rise to a generalized conic structure, as shown by a low dimensional illustration in Figure 2.8.

A desired outcome of locally linear models is interpretability. Despite their success, these deep neural net denoisers are mysterious: we lack both intuition and formal understanding of the mechanisms they implement and most importantly what priors are being learned by these models. The proposed bias-free architecture lends itself to local linear algebraic analysis through the Jacobian of the denoising map, which provides high level interpretability. The analysis reveals locally adaptive properties of the learned models, akin to existing nonlinear filtering algorithms. In addition, we show that the deep networks implicitly perform a projection onto an adaptively-selected low-dimensional subspace capturing features of natural images.

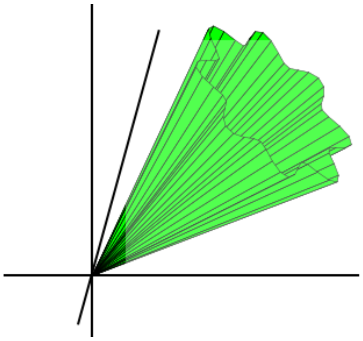


Figure 2.8: Generalized conic manifold

2.3.1 Network Bias Impairs Generalization

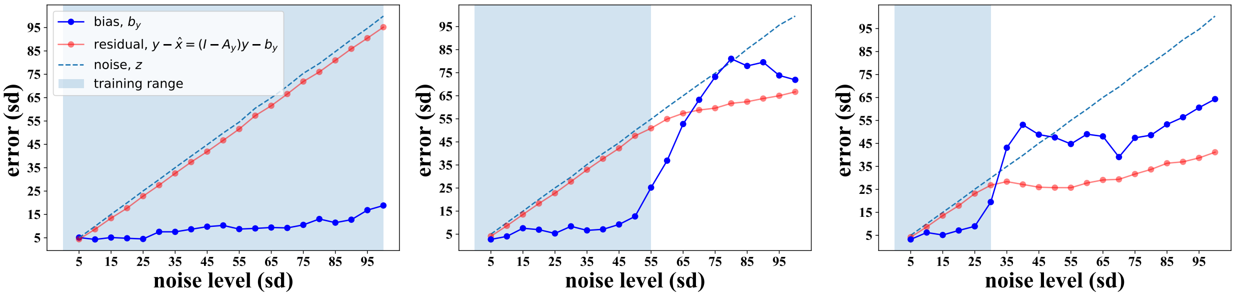
We assume a measurement model in which images are corrupted by additive noise: $y = x + n$, where $x \in \mathbb{R}^N$ is the original image, containing N pixels, n is an image of i.i.d. samples of Gaussian noise with variance σ^2, and y is the noisy observation. The denoising problem consists of finding a function $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$, that provides a good estimate of the original image, x. Commonly, one minimizes the mean squared error : $f = \arg\min_g E||x - g(y)||^2$, where the expectation is taken over some distribution over images, x, as well as over the distribution of noise realizations. In deep learning, the denoising function g is parameterized by the weights of the network, so the optimization is over these parameters. If the noise standard deviation, σ, is unknown, the expectation must also be taken over a distribution of σ. This problem is often called *blind denoising* in the literature. Here, we study the generalization performance of CNNs *across* noise levels σ, i.e. when they are tested on noise levels not included in the training set.

Feedforward neural networks with rectified linear units (ReLUs) are piecewise affine: for a given activation pattern of the ReLUs, the effect of the network on the input is a cascade of linear transformations (convolutional or fully connected layers, W_k), additive constants (b_k), and pointwise multiplications by a binary mask corresponding to the fixed activation pattern (R). Since each of these is affine, the entire cascade implements a single affine transformation. For a fixed noisy input image $y \in \mathbb{R}^N$ with N pixels, the function $f : \mathbb{R}^N \to \mathbb{R}^N$ computed by a denoising neural network may be written

$$f(y) = W_L R(W_{L-1} ... R(W_1 y + b_1) + ... b_{L-1}) + b_L = A_y y + b_y, \tag{2.7}$$

where $A_y \in \mathbb{R}^{N \times N}$ is the Jacobian of $f(\cdot)$ evaluated at input y, and $b_y \in \mathbb{R}^N$ represents the *net bias*. The subscripts on A_y and b_y serve as a reminder that both depend on the ReLU activation patterns, which in turn depend on the input vector y.

Based on equation 2.7 we can perform a first-order decomposition of the error or *residual* of the neural network for a specific input: $y - f(y) = (I - A_y)y - b_y$. Figure 2.9 shows the magnitude of the residual and the constant, which is equal to the net bias b_y, for a range of noise levels. Over the training range, the net bias is small, implying that the linear term is responsible for most of the denoising (see Figures 2.9 for a visualization of both components). However, when the network is evaluated at noise levels outside of the training range, the norm of the bias increases dramatically, and the residual is significantly smaller than the noise, suggesting a form of overfitting. Indeed, network performance generalizes very poorly to noise levels outside the training range. This is illustrated for an example image in Figure 2.10, and demonstrated through extensive experiments in Section 2.3.3.

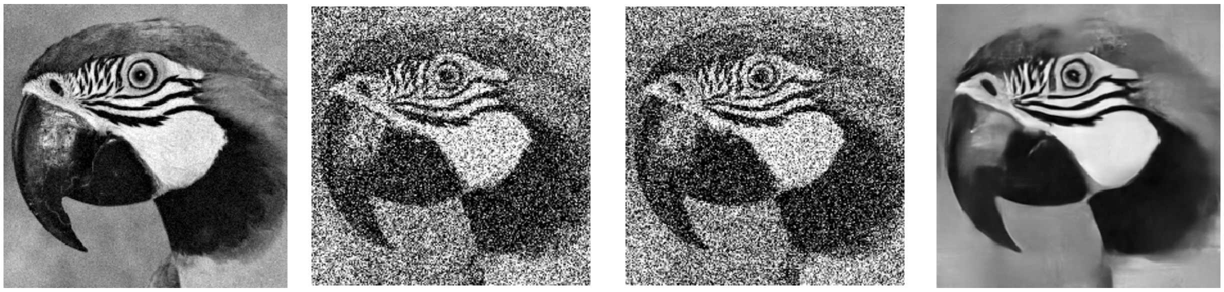


(a)

(b)

(c)

Figure 2.9: Bias component of the locally affine denoising mapping. First-order analysis of the residual of a denoising convolutional neural network as a function of noise level. The plots show the norms of the residual and the net bias averaged over 100 20 × 20 natural-image patches for networks trained over different training ranges. The range of noises used for training is highlighted in blue. **(a)** When the network is trained over the full range of noise levels ($\sigma \in [0, 100]$) the net bias is small, growing slightly as the noise increases. **(b-c)** When the network is trained over the a smaller range ($\sigma \in [0, 55]$ and $\sigma \in [0, 30]$), the net bias grows explosively for noise levels beyond the training range. This coincides with a dramatic drop in performance, reflected in the difference between the magnitudes of the residual and the true noise. The CNN used for this example is DnCNN (Zhang et al., 2017a); using alternative architectures yields similar results as shown in Figure A.1.



Noisy training image, $\sigma = 10$ (max level)

Noisy test image, $\sigma = 90$

Test image, denoised by CNN

Test image, denoised by BF-CNN

Figure 2.10: Effect of net bias on generalization for an example image. Denoising of an example natural image by a CNN and its bias-free counterpart (BF-CNN), both trained over noise levels in the range $\sigma \in [0, 10]$ (image intensities are in the range $[0, 255]$). The CNN performs poorly at high noise levels ($\sigma = 90$, far beyond the training range), whereas BF-CNN performs at state-of-the-art levels. The CNN used for this example is DnCNN (Zhang et al., 2017a); using alternative architectures yields similar results (see Section 2.3.3).

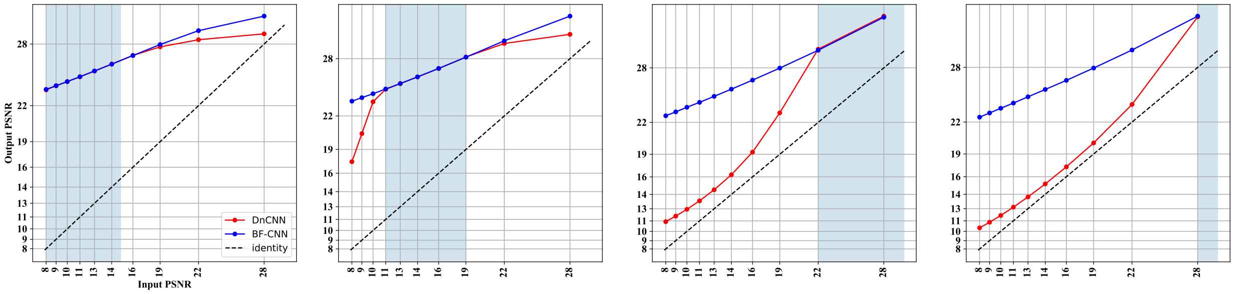


Figure 2.11: Effect of net bias on generalization for a test dataset. Comparison of the performance of a CNN and a BF-CNN with the same architecture for the experimental design described in Section 2.3.3. The performance is quantified by the PSNR of the denoised image as a function of the input PSNR. Both networks are trained over a fixed ranges of noise levels indicated by a blue background. In all cases, the performance of BF-CNN generalizes robustly beyond the training range, while that of the CNN degrades significantly. The CNN used for this example is DnCNN (Zhang et al., 2017a); using alternative architectures yields similar results (see Figures A.2 and A.3).

2.3.2 Introducing Bias-Free Networks

We showed that CNNs overfit to the noise levels present in the training set, and that this is associated with wild fluctuations of the net bias b_y. This suggests that the overfitting might be ameliorated by removing additive (bias) terms from every stage of the network, resulting in a *bias-free* CNN (BF-CNN). Note that bias terms are also removed from the batch-normalization used during training. This simple change in the architecture has an interesting consequence. If the CNN has ReLU activations the denoising map is locally homogeneous, and consequently *invariant to scaling*: rescaling the input by a constant value simply rescales the output by the same amount, just as it would for a linear system.

Lemma 2.1. *Let $f_{\mathrm{BF}} : \mathbb{R}^N \to \mathbb{R}^N$ be a feedforward neural network with ReLU activation functions and no additive constant terms in any layer. For any input $y \in \mathbb{R}$ and any nonnegative constant α,*

$$f_{\mathrm{BF}}(\alpha y) = \alpha f_{\mathrm{BF}}(y). \tag{2.8}$$

Proof. We can write the action of a bias-free neural network with L layers in terms of the weight matrix W_i, $1 \leq i \leq L$, of each layer and a rectifying operator $\mathcal{R}$, which sets to zero any negative entries in its input. Multiplying by a nonnegative constant does not change the sign of the entries of a vector, so for any z with the right dimension and any $\alpha > 0$ $\mathcal{R}(\alpha z) = \alpha\mathcal{R}(z)$, which implies

$$f_{\text{BF}}(\alpha y) = W_L\mathcal{R}(W_{L-1}\cdots\mathcal{R}(W_1\alpha y)) = \alpha W_L\mathcal{R}(W_{L-1}\cdots\mathcal{R}(W_1 y)) = \alpha f_{\text{BF}}(y). \tag{2.9}$$

□

Note that networks with nonzero net bias are not scaling invariant because scaling the input may change the activation pattern of the ReLUs. Scaling invariance is intuitively desireable for a denoising method operating on natural images; a rescaled natural image is still a natural image. Note that Lemma 2.1 holds for networks with skip connections where the feature maps are concatenated or added, because both of these operations are linear.

In the following sections we demonstrate that removing all additive terms in CNN architectures has two important consequences: (1) the networks gain the ability to generalize to noise levels not encountered during training (as illustrated by Figure 2.10 the improvement is striking), and (2) the denoising mechanism can be analyzed locally via linear-algebraic tools that reveal intriguing ties to more traditional denoising methodology such as nonlinear filtering and sparsity-based techniques.

2.3.3 Bias-free Networks Generalize Across Noise Levels

In order to evaluate the effect of removing the net bias in denoising CNNs, we compare several popular architectures to their bias-free counterparts, which are exactly the same except for the absence of any additive constants within the networks (note that this includes the batch-normalization additive parameter). These architectures include popular features of existing neural-network techniques in image processing: convolutional filters, downsampling, and skip connections. More specifically, we examine the following models (see Section A.1 for additional details):

- DnCNN (Zhang et al., 2017a): A feedforward CNN with 20 convolutional layers, each consisting of 3 × 3 filters, 64 channels, batch normalization (Ioffe and Szegedy, 2015), a ReLU nonlinearity, and a skip connection from the initial layer to the final layer.
- Recurrent CNN: A recurrent architecture inspired by (Zhang et al., 2018a) where the basic module is a CNN with 5 layers, 3 × 3 filters and 64 channels in the intermediate layers. The order of the recurrence is 4.
- UNet (Ronneberger et al., 2015a): A multiscale architecture with 9 convolutional layers and skip connections between the different scales.
- Simplified DenseNet: CNN with skip connections inspired by the DenseNet architecture (Huang et al., 2017; Zhang et al., 2018b).

We train each network to denoise images corrupted by i.i.d. Gaussian noise over a range of standard deviations (the *training range* of the network). We then evaluate the network for noise levels that are both within and beyond the training range. Our experiments are carried out on 180 × 180 natural images from the Berkeley Segmentation Dataset (Martin et al., 2001) to be consistent with previous results (Chen and Pock, 2017; Schmidt and Roth, 2014; Zhang et al., 2017a). Additional details about the dataset and training procedure are provided in Section A.2.

Figures 2.11 and A.2 show our results. For a wide range of different training ranges, and for all architectures, we observe the same phenomenon: the performance of CNNs is good over the training range, but degrades dramatically at new noise levels; in stark contrast, the corresponding BF-CNNs provide strong denoising performance over noise levels outside the training range. Figure 2.10 shows an example image, demonstrating visually the striking difference in generalization performance between a CNN and its corresponding BF-CNN. Our results provide strong evidence that removing net bias in CNN architectures results in effective generalization to noise levels out of the training range.

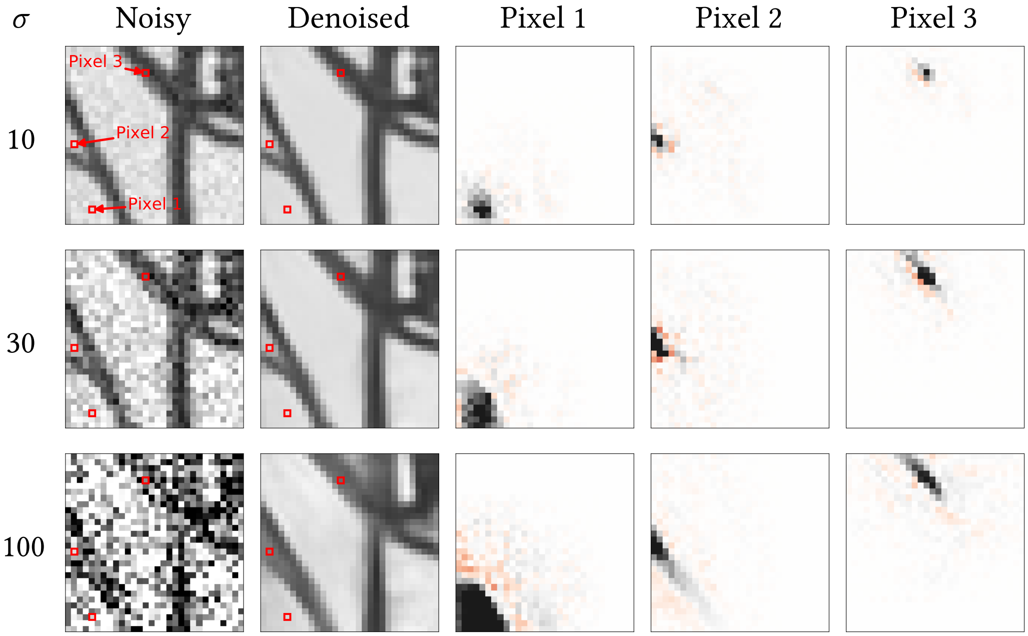


Figure 2.12: Visualization of the linear weighting functions (rows of A_y in equation 2.10) of a BF-CNN for three example pixels of an input image, and three levels of noise. The images in the three rightmost columns show the weighting functions used to compute each of the indicated pixels (red squares). All weighting functions sum to one, and thus compute a local average (note that some weights are negative, indicated in red). Their shapes vary substantially, and are adapted to the underlying image content. As the noise level σ increases, the spatial extent of the weight functions increases in order to average out the noise, while respecting boundaries between different regions in the image, which results in dramatically different functions for each pixel. The CNN used for this example is DnCNN (Zhang et al., 2017a).

2.3.4 Revealing the Denoising Mechanisms Learned by BF-CNNs

In this section we perform a local analysis of BF-CNN networks, which reveals the underlying denoising mechanisms learned from the data. A bias-free network is strictly linear, and its net action can be expressed as

$$f_{\mathrm{BF}}(y) = W_L R(W_{L-1} ... R(W_1 y)) = A_y y, \tag{2.10}$$

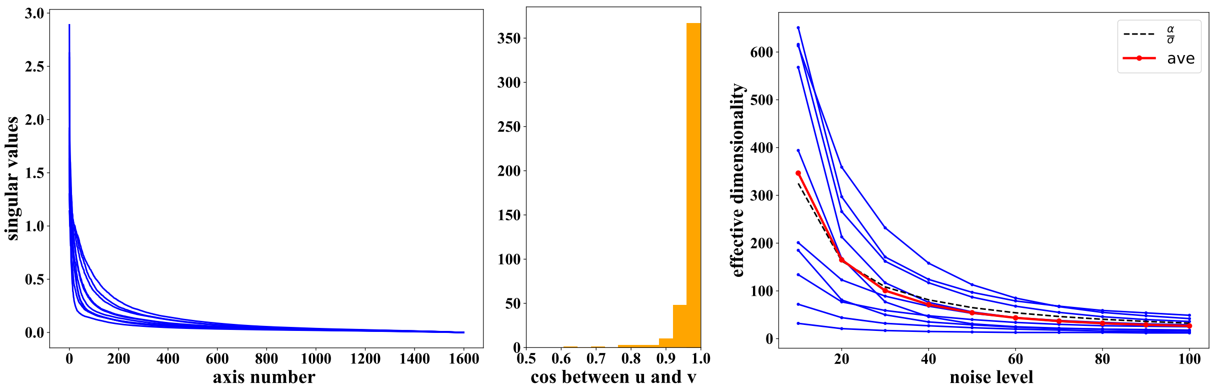
where A_y is the Jacobian of $f_{\mathrm{BF}}(\cdot)$ evaluated at y. The Jacobian at a fixed input provides a local characterization of the denoising map. In order to study the map we perform a linear-algebraic analysis of the Jacobian. Our approach is similar in spirit to visualization approaches– proposed in the context of image classification– that differentiate neural-network functions with respect to their input (e.g. (Montavon et al., 2017; Simonyan et al., 2013)).

2.3.4.1 Nonlinear adaptive filtering

The linear representation of the denoising map given by equation 2.10 implies that the ith pixel of the output image is computed as an inner product between the ith row of A_y, denoted $a_y(i)$, and the input image:

$$f_{\mathrm{BF}}(y)(i) = \sum_{j=1}^{N} A_y(i,j) y(j) = a_y(i)^T y. \tag{2.11}$$

The vectors $a_y(i)$ can be interpreted as *adaptive filters* that produce an estimate of the denoised pixel via a weighted average of noisy pixels. Examination of these filters reveals their diversity, and their relationship to the underlying image content: they are adapted to the local features of the noisy image, averaging over homogeneous regions of the image without blurring across edges. This is shown for two separate examples and a range of noise levels in Figures 2.12. We observe that the equivalent filters of all architectures adapt to image structure.



(a) (b) (c)

Figure 2.13: Analysis of the SVD of the Jacobian of a BF-CNN for ten natural images, corrupted by noise of standard deviation $\sigma = 50$. **(a)** Singular value distributions. For all images, a large proportion of the values are near zero, indicating (approximately) a projection onto a subspace (the *signal subspace*). **(b)** Histogram of dot products (cosine of angle) between the left and right singular vectors that lie within the signal subspaces. **(c)** Effective dimensionality of the signal subspaces (computed as sum of squared singular values) as a function of noise level. For comparison, the total dimensionality of the space is 1600 (40 × 40 pixels). Average dimensionality (red curve) falls approximately as the inverse of σ (dashed curve). The CNN used for this example is DnCNN (Zhang et al., 2017a); using alternative architectures yields similar results (see Figure A.5).

Classical Wiener filtering (Wiener, 1950) denoises images by computing a local average dependent on the noise level. As the noise level increases, the averaging is carried out over a larger region. As illustrated by Figures 2.12, the equivalent filters of BF-CNNs also display this behavior. The crucial difference is that the filters are adaptive. The BF-CNNs learn such filters implicitly from the data, in the spirit of modern nonlinear spatially-varying filtering techniques designed to preserve fine-scale details such as edges (e.g. (Tomasi and Manduchi, 1998), see also (Milanfar, 2012a) for a comprehensive review, and (Choi et al., 2018) for a recent learning-based approach).

2.3.5 Projection onto adaptive low-dimensional subspaces

The local linear structure of a BF-CNN facilitates analysis of its functional capabilities via the singular value decomposition (SVD). For a given input y, we compute the SVD of the Jacobian matrix: $A_y = USV^T$, with U and V orthogonal matrices, and S a diagonal matrix. We can decompose the effect of the network on its input in terms of the left singular vectors $\{U_1, U_2 \ldots, U_N\}$ (columns of U), the singular values $\{s_1, s_2 \ldots, s_N\}$ (diagonal elements of S), and the right singular vectors $\{V_1, V_2, \ldots V_N\}$ (columns of V):

$$f_{\text{BF}}(y) = A_y y = USV^T y = \sum_{i=1}^{N} s_i (V_i^T y) U_i. \tag{2.12}$$

The output is a linear combination of the left singular vectors, each weighted by the projection of the input onto the corresponding right singular vector, and scaled by the corresponding singular value.

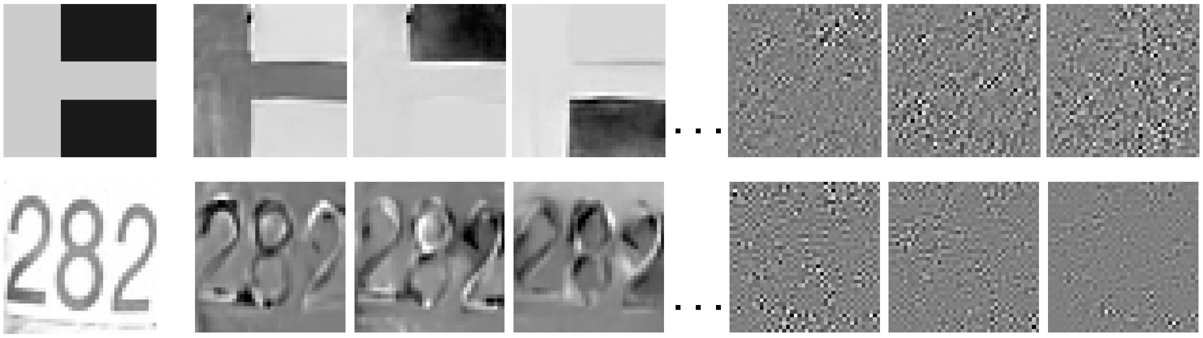


Figure 2.14: Visualization of left singular vectors of the Jacobian of a BF-CNN, evaluated on two different images (top and bottom rows), corrupted by noise with standard deviation $\sigma = 50$. The left column shows original (clean) images. The next three columns show singular vectors corresponding to non-negligible singular values. The vectors capture features from the clean image. The last three columns on the right show singular vectors corresponding to singular values that are almost equal to zero. These vectors are noisy and unstructured. The CNN used for this example is DnCNN (Zhang et al., 2017a); using alternative architectures yields similar results (see Figure A.4).

Analyzing the SVD of a BF-CNN on a set of ten natural images reveals that most singular values are very close to zero (Figure 2.13a). The network is thus discarding all but a very low-dimensional portion of the input image. We also observe that the left and right singular vectors corresponding to the singular values with non-negligible amplitudes are approximately the same (Figure 2.13b). This means that the Jacobian is (approximately) symmetric, and we can interpret the action of the network as projecting the noisy signal onto a low-dimensional subspace. This is confirmed by visualizing the singular vectors as images (Figure 2.14). The singular vectors corresponding to non-negligible singular values are seen to capture features of the input image; those corresponding to near-zero singular values are unstructured. The BF-CNN therefore implements an approximate projection onto an adaptive *signal subspace* that preserves image structure, while suppressing the noise.

This is similar to wavelet thresholding schemes. From a linear-algebraic perspective, these algorithms operate by projecting the noisy input onto a lower-dimensional subspace that contains plausible signal content. The projection eliminates the orthogonal complement of the subspace, which mostly contains noise. The advantage of data-driven deep-learning models is that in addition to the thresholding values, the image subspace is also adaptive to the underlying image.

We can define an "effective dimensionality" of the signal subspace as $d := \sum_{i=1}^{N} s_i^2$, the amount of variance captured by applying the linear map to an N-dimensional Gaussian noise vector with variance σ^2, normalized by the noise variance. The remaining variance equals

$$E_n||A_y n||^2 = E_n||U_y S_y V_y^T n||^2 = E_n||S_y n||^2 = E_n \sum_{i=1}^{N} s_i^2 n_i^2 = \sum_{i=1}^{N} s_i^2 E_n(n_i^2) \approx \sigma^2 \sum_{i=1}^{N} s_i^2,$$

where E_n indicates expectation over noise n, so that $d = E_n||A_y n||^2/\sigma^2 = \sum_{i=1}^{N} s_i^2$.

When we examine the preserved signal subspace, we find that the clean image lies almost completely within it. For inputs of the form $y := x + n$ (where x is the clean image and n the noise), we find that the subspace spanned by the singular vectors up to dimension d contains x almost entirely, in the sense that projecting x onto the subspace preserves most of its energy. This holds for the whole range of noise levels over which the network is trained (Figure 2.15).

We also find that for any given clean image, the effective dimensionality of the signal subspace (d) decreases systematically with noise level (Figure 2.13c). At lower noise levels the network detects a richer set of image features, and constructs a larger signal subspace to capture and preserve them. Empirically, we found that (on average) d is approximately proportional to $\frac{1}{\sigma}$ (see dashed line in Figure 2.13c). These signal subspaces are nested: the subspaces corresponding to

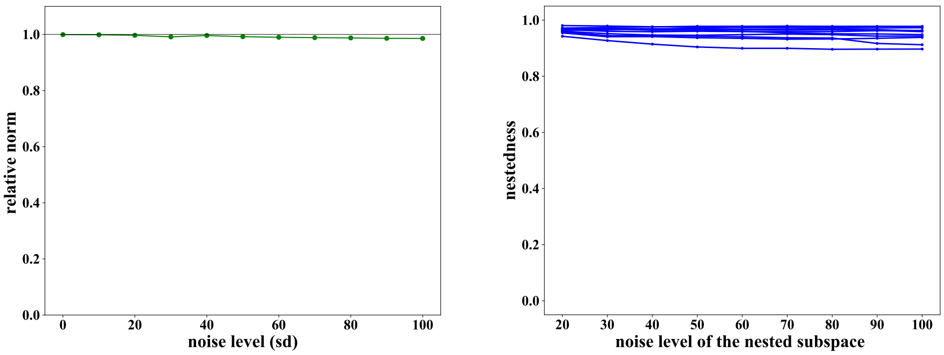


Figure 2.15: Signal subspace properties. **Left:** Signal subspace, computed from Jacobian of a BF-CNN evaluated at a particular noise level, contains the clean image. Specifically, the fraction of squared ℓ_2 norm preserved by projection onto the subspace is nearly one as σ grows from 10 to 100 (relative to the image pixels, which lie in the range $[0, 255]$). Results are averaged over 50 example clean images. **Right:** Signal subspaces at different noise levels are nested. The subspace axes for a higher noise level lie largely within the subspace obtained for the lowest noise level ($\sigma = 10$), as measured by the sum of squares of their projected norms. Results are shown for 10 example clean images.

lower noise levels contain more than 95% of the subspace axes corresponding to higher noise levels (Figure 2.15).

Finally, we note that this behavior of the signal subspace dimensionality, combined with the fact that it contains the clean image, explains the observed denoising performance across different noise levels (Figure 2.11). Specifically, if we assume $d \approx \alpha/\sigma$, the mean squared error is proportional to σ:

$$\begin{aligned} \text{MSE} &= E_n||A_y(x+n) - x||^2 \\ &\approx E_n||A_y n||^2 \\ &\approx \sigma^2 d \\ &\approx \alpha\,\sigma \end{aligned} \tag{2.13}$$

Note that this result runs contrary to the intuitive expectation that MSE should be proportional to the noise variance, which would be the case if the denoiser operated by projecting onto a fixed subspace. The scaling of MSE with the square root of the noise variance implies that the PSNR of the denoised image should be a linear function of the input PSNR, with a slope of $1/2$, consistent with the empirical results shown in Figure 2.11. Note that this behavior holds even when the networks are trained only on modest levels of noise (e.g., $\sigma \in [0, 10]$).

2.4 Discussion

In the first half of this chapter, we describe a framework for extracting the prior embedded in a denoiser. Specifically, we develope a stochastic iterative score ascent (SISA) algorithm that uses the denoiser to draw high-probability samples from its implicit prior. The derivation is based on a simple and direct expression relating denoising to priors, and a few basic empirical facts about universal CNN denoisers. We empirically demonstrated its efficiency in Fig. 2.2. It is worth noting that the assumed use of Gaussian additive noise and MSE objective in training the denoiser is necessary only to justify the use of Miyasawa's expression (Eq. 2.3), but does not impose any such restrictions on use of the trained denoiser for sampling from the implicit prior. Our method for image generation offers a means of visualizing the implicit prior of a denoiser, which arises from the combination of architecture, optimization, regularization, and training set. As such, it might offer a means of experimentally isolating and elucidating the effects of these components. Although the method can be used with any universal least-squares denoiser designed or trained to remove Gaussian noise, the complexity of the embedded prior, and thus the quality of samples, relies heavily on the expressive power of the denoiser, as well as the diversity of the training set.

In the second half of this chapter, we discuss properties of CNN denoisers at the heart of our synthesis algorithm. We show that removing constant terms from CNN architectures ensures strong generalization across noise levels, and also provides interpretability of the denoising method via linear-algebra techniques. We provide insights into the relationship between bias and generalization through a set of observations. Theoretically, we argue that if the denoising network operates by projecting the noisy observation onto a linear space of "clean" images, then that space should include all rescalings of those images, and thus, the origin. This property can be guaranteed by eliminating bias from the network. Empirically, in networks that allow bias, the net bias of the trained network is quite small within the training range. However, outside the training range the net bias grows dramatically resulting in poor performance, which suggests that the bias may be the cause of the failure to generalize. In addition, when we remove bias from the architecture, we preserve performance within the training range, but achieve near-perfect generalization, even to noise levels more than 10x those in the training range. These observations do not fully elucidate how our network achieves its remarkable generalization- only that bias prevents that generalization, and its removal allows it.

It is of interest to examine whether bias removal can facilitate generalization in noise distributions beyond Gaussian, as well as other image-processing tasks, such as image restoration and image compression. We have trained bias-free networks on uniform noise and found that they generalize outside the training range. In fact, bias-free networks trained for Gaussian noise generalize well when tested on uniform noise.

Finally, our linear-algebraic analysis uncovers interesting aspects of the denoising map, but these interpretations are very local: small changes in the input image change the activation patterns of the network, resulting in a change in the corresponding linear mapping. Extending the analysis to reveal global characteristics of the neural-network functionality is a challenging direction for future research.

3 Generalization beyond memorizing training images

Diffusion methods have demonstrated ever-more impressive capabilities for sampling from high-dimensional image densities (Ho et al., 2020; Kadkhodaie and Simoncelli, 2020; Song and Ermon, 2019a). However, approximating a continuous density in a high-dimensional space is notoriously difficult: do these networks actually achieve this feat, learning from a relatively small training set to generate high-quality samples, in apparent defiance of the curse of dimensionality? If so, this must be due to their inductive biases, that is, the restrictions that the architecture and optimization place on the learned denoising function. But the approximation class associated with these models is not well understood. Here, we take several steps toward elucidating this mystery.

Several recently reported results show that, when the training set is small relative to the network capacity, diffusion generative models do not approximate a continuous density, but rather memorize samples of the training set, which are then reproduced (or recombined) when generating new samples (Carlini et al., 2023; Somepalli et al., 2023) — see Figure 3.1. This is a form of overfitting (high model variance). Here, we confirm this behavior for DNNs trained on small data sets, but demonstrate that these same models do not memorize when trained on sufficiently

This chapter is published (Kadkhodaie et al., 2023b) in collaboration with Florentin Guth and Stephane Mallat. The code for generating results is available at `https://github.com/LabForComputationalVision/memorization_generalization_in_diffusion_models`

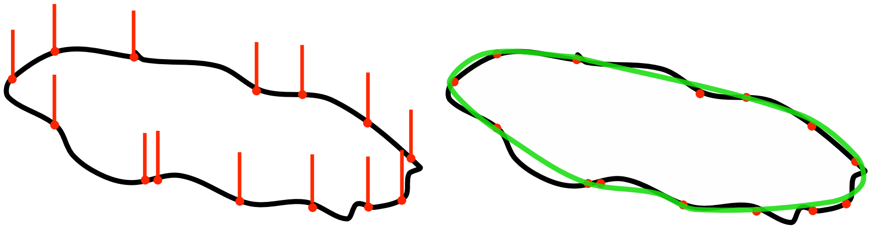


Figure 3.1: Memorization vs. generalization in 2D. Approximating the true continuous underlying distribution defined over a one-dimensional manifold can lead to one of the two extreme cases or anything in between. On the left, the approximated density is the empirical density; a set of delta functions corresponding to the training examples. This model is discrete, and samples drawn from it are high quality, because they all lie on the true manifold. However, diversity of samples will be low. On the right, the learned density is defined over a continuous manifold approximating the trough manifold. Samples from this density are novel and diverse. If the estimator is low biased, then the samples drawn from this density will be high quality because they lie on or near the true manifold. Note that, assuming a fixed model bias, going from memorization to generalization reduces the dependence of the learned density on the individual examples in the training set.

large sets. Specifically, we show that two denoisers trained on sufficiently large non-overlapping sets converge to essentially the same denoising function. That is, the learned model becomes independent of the training set (i.e., model variance falls to zero). As a result, when used for image generation, these networks produce nearly identical samples. These results provide stronger and more direct evidence of generalization than standard comparisons of average performance on train and test sets. This generalization can be achieved with large but realizable training sets (for our examples, roughly 10^5 images suffices), reflecting powerful inductive biases of these networks. Moreover, sampling from these models produces images of high visual quality, implying that these inductive biases are well-matched to the underlying distribution of photographic images (Goyal and Bengio, 2022; Griffiths et al., 2023; Wilson and Izmailov, 2020).

To study these inductive biases, we exploit the relationship between denoising and density estimation. In Section 2.3, we described the operation of the DNN denoiser as an approximate projection onto an image subspace. In this chapter, using the same linear algebraic tools, we expand our interpretation of the denoising operation from an approximate projection to a shrinkage operation. We find that DNN denoisers trained on photographic images perform a shrinkage operation in an orthonormal basis consisting of harmonic functions that are adapted to the geometry of features in the underlying image. We refer to these as *geometry-adaptive harmonic bases* (GAHBs). This observation, taken together with the generalization performance of DNN denoisers, suggests that optimal bases for denoising photographic images are GAHBs and, moreover, that inductive biases of DNN denoisers encourage such bases. To test this more directly, we examine a particular class of images whose intensity variations are regular over regions separated by regular contours. A particular type of GAHB, known as “bandlets” (Peyré and Mallat, 2008), have been shown to be near-optimal for denoising these images (Dossal et al., 2011). We observe that the DNN denoiser operates within a GAHB similar to a bandlet basis, also achieving near-optimal performance. Thus the inductive bias enables the network to appropriately estimate the score in these cases.

If DNN denoisers induce biases towards the GAHB approximation class, then they should perform sub-optimally for distributions whose optimal bases are not GAHBs. To investigate this, we train DNN denoisers on image classes supported on low-dimensional manifolds, for which the optimal denoising basis is only partially constrained. Specifically, an optimal denoiser (for small noise) should project a noisy image on the tangent space of the manifold. We observe that the DNN denoiser closely approximates this projection, but also partially retains content lying within a subspace spanned by a set of additional GAHB vectors. These suboptimal components reflect the GAHB inductive bias.

3.1 Diffusion model variance and denoising generalization

Diffusion models learn the scores of the distributions of noise-corrupted images, rather than approximating this density directly. Here, we show that the denoising error provides a bound on the density modeling error, and use this to analyze the convergence of the density model.

3.1.1 Diffusion models and denoising

As discussed in Section 2.1, The density $p_\sigma(y)$ of noisy images is then related to $p(x)$ through marginalization over x:

$$p_\sigma(y) = \int p(y|x)\, p(x)\, \mathrm{d}x = \int g_\sigma(y - x)\, p(x)\, \mathrm{d}x,$$

where $g_\sigma(z)$ is the density of z. Hence, $p_\sigma(y)$ is obtained by convolving $p(x)$ with a Gaussian with standard deviation σ. The family of densities $\{p_\sigma(y); \sigma \geq 0\}$ forms a scale-space representation of $p(x)$, analogous to the temporal evolution of a diffusion process.

Diffusion models learn an approximation $s_\theta(y)$ (dropping the σ dependence for simplicity) of the scores $\nabla \log p_\sigma(y)$ of the blurred densities $p_\sigma(y)$ at all noise levels σ. The collection of these score models implicitly defines a model $p_\theta(x)$ of the density of clean images $p(x)$ through a reverse diffusion process. The error of the generative model, as measured by the KL divergence between $p(x)$ and $p_\theta(x)$, is then controlled by the integrated score error across all noise levels (Song et al., 2021a):

$$\mathrm{KL}\left(p(x) \parallel p_\theta(x)\right) \leq \int_0^\infty \mathop{\mathbb{E}}_{y}\left[\|\nabla \log p_\sigma(y) - s_\theta(y)\|^2\right] \sigma\, \mathrm{d}\sigma. \tag{3.1}$$

The key to learning the scores is Miyasawa/Tweedie equation 2.3 that relates them to the mean of the corresponding posteriors:

$$\nabla \log p_\sigma(y) = (\mathop{\mathbb{E}}_{x}[x \,|\, y] - y)/\sigma^2.$$

The score is learned by training a denoiser $f_\theta(y)$ to minimize the mean squared error (MSE)

(Raphan and Simoncelli, 2011; Vincent, 2011b):

$$\mathrm{MSE}(f_\theta, \sigma^2) = \mathop{\mathbb{E}}_{x,y}\left[\|x - f_\theta(y)\|^2\right], \tag{3.2}$$

so that $f_\theta(y) \approx \mathbb{E}_x\left[x \,|\, y\right]$. This estimated conditional mean is used to recover the estimated score using eq. (2.3): $s_\theta(y) = (f_\theta(y) - y)/\sigma^2$. As we show in Appendix B.3.2, the error in estimating the density $p(x)$ is bounded by the integrated optimality gap of the denoiser across noise levels:

$$\mathrm{KL}\left(p(x) \parallel p_\theta(x)\right) \leq \int_0^\infty \left(\mathrm{MSE}(f_\theta, \sigma^2) - \mathrm{MSE}(f^\star, \sigma^2)\right)\, \sigma^{-3}\, \mathrm{d}\sigma, \tag{3.3}$$

where $f^\star(y) = \mathbb{E}_x\left[x \,|\, y\right]$ is the optimal denoiser. Thus, learning the true density model is equivalent to performing optimal denoising at all noise levels. Conversely, a suboptimal denoiser introduces a score approximation error, which in turn can result in an error in the modeled density.

Generally, the optimal denoising function $f^\star$ (as well as the "true" distribution, $p(x)$) is unknown for photographic images, which makes numerical evaluation of sub-optimality challenging. We can however separate deviations from optimality arising from model bias and model variance. Model variance measures the size of the approximation class, and hence the strength (or restrictiveness) of the inductive biases. It can be evaluated without knowledge of $f^\star$. Here, we define generalization as near-zero model variance (i.e., an absence of overfitting), which is agnostic to model bias. This is the subject of Section 3.1.2. Model bias measures the distance of the true score to the approximation class, and thus the alignment between the inductive biases and the data distribution. In the context of photographic images, visual quality of generated samples can be a qualitative indicator of the model bias, although high visual quality does not necessarily guarantee low model bias. We evaluate model bias in Section 3.2.2 by considering synthetic image classes for which $f^\star$ is approximately known.

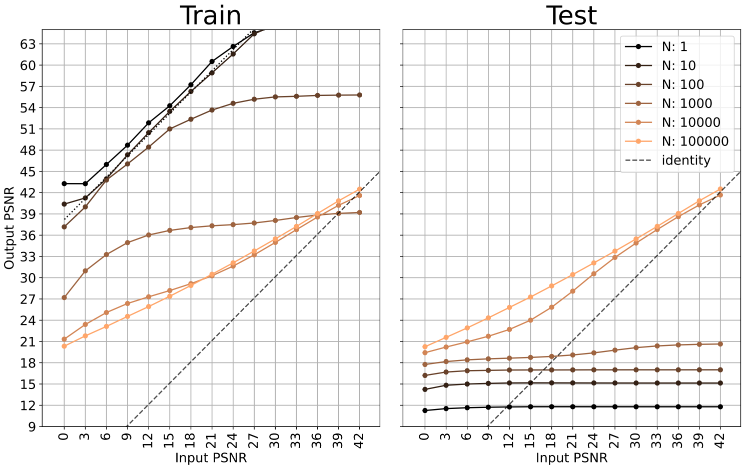


Figure 3.2: Transition from memorization to generalization, for a UNet denoiser trained on face images. Each curve shows the denoising error (output PSNR, ten times log10 ratio of squared dynamic range to MSE) as a function of noise level (input PSNR), for a training set of size N. As N increases, performance on the training set generally worsens (left), while performance on the test set improves (right). For $N = 1$ and $N = 10$, the train PSNR improves with unit slope, while test PSNR is poor, independent of noise level, a sign of memorization. The increase in test performance on small noise levels at $N = 1000$ is indicative of the transition phase from memorization to generalization. At $N = 10^5$, test and train PSNR are essentially identical, and the model is no longer overfitting the training data.

3.1.2 Transition from memorization to generalization

DNNs are susceptible to overfitting, because the number of training examples is typically small relative to the model capacity. Since density estimation, in particular, suffers from the curse of dimensionality, overfitting is of more concern in the context of generative models. An overfitted denoiser performs well on training images but fails to generalize to test images, resulting in low-diversity generated images. Consistent with this, several papers have reported that diffusion models can memorize their training data (Carlini et al., 2023; Dar et al., 2023; Somepalli et al., 2023; Zhang et al., 2023). To directly assess this, we compared denoising performance on training and test data for different training set sizes N. We trained denoisers on subsets of the (downsampled) CelebA dataset (Liu et al., 2015) of size $N = 10^0, 10^1, 10^2, 10^3, 10^4, 10^5$. We used a UNet architecture (Ronneberger et al., 2015b), which is composed of 3 convolutional encoder and decoder blocks with rectifying non-linearities. These denoisers are universal and blind: they operate on all noise levels without having noise level as an input (Mohan* et al., 2020). Networks are trained to minimize mean squared error (3.2). See Appendix B.1 for architecture and training details.

Results are shown in Figure 3.2. When $N = 1$, the denoiser essentially memorizes the single training image, leading to a high test error. Increasing N substantially increases the performance on the test set while worsening performance on the training set, as the network transitions from memorization to generalization. At $N = 10^5$, empirical test and train error are matched for all noise levels.

To investigate this generalization further, we train denoisers on *non-overlapping* subsets of CelebA of various size N (see Figure 3.4). We then generate samples using the scores learned by each denoiser, through the SISA algorithm discussed in Chapter 2. Figure 3.3 shows samples generated by these denoisers, initialized from the same noise sample. For small N, the networks memorize their respective training images. However, for large N, the networks converge to the same score function (and thus sample from the same model density), generating nearly identical samples. The bottom portion of figure compare the similarity between pairs of images generated by the two networks with the similarity between each generated sample and the most similar image in the corresponding training set. For small N, a significant number of samples are perfectly matched to training set images, and the samples from the two networks are not matched. With increasing N, images drawn from the two denoisers become more similar to each other, and less similar to the closest image in their respective training sets, demonstrating the convergence to a generalized common distribution. This surprising behavior provides a much stronger demonstration of convergence than comparison of average train and test performance. More examples of pairs of images synthesized at convergence are shown in Figure 3.5. To see an example of less similar pairs of synthesized images see Figure 3.6.

Convergence of model variance is robust to the change of data distribution and architecture. Figure 3.7 shows convergence for models trained on LSUN bedrooms dataset. Figures 3.8, 3.9, 3.10 and 3.11 show convergence for BF-CNN models trained on CelebA and LSUN bedrooms datasets.

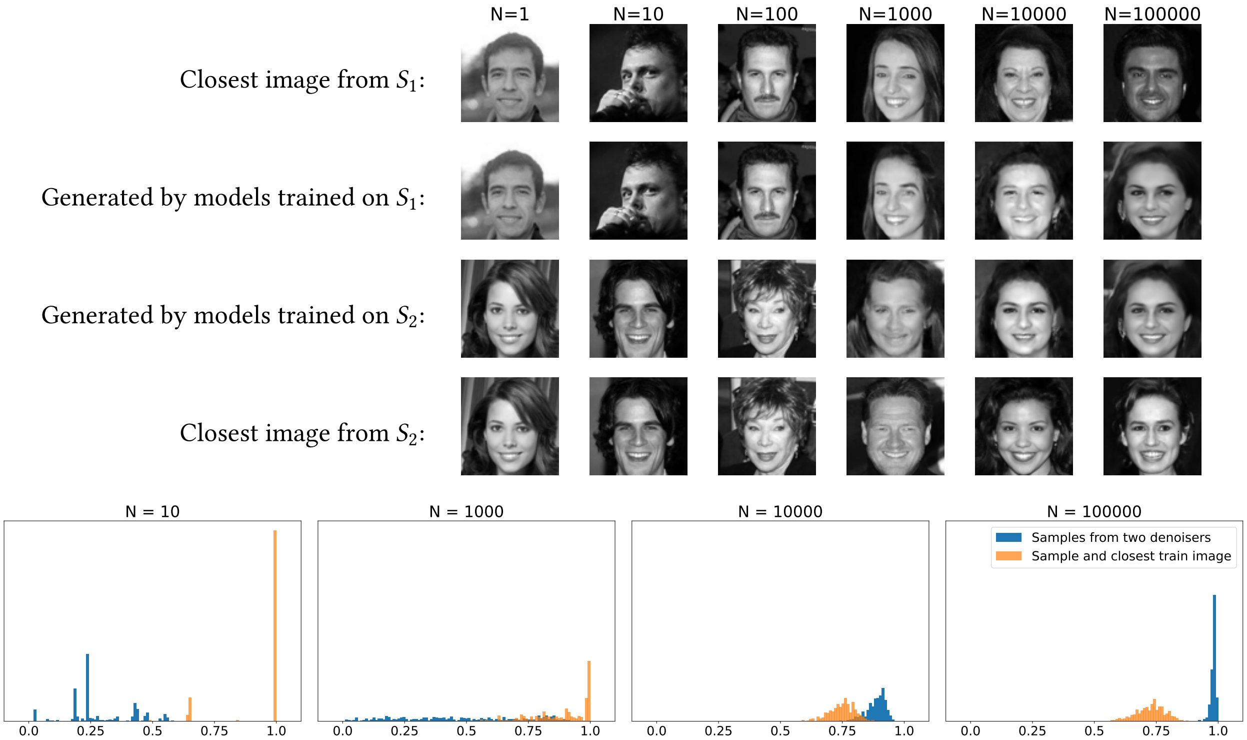


Figure 3.3: Convergence of model variance. Diffusion models are trained on non-overlapping subsets S_1 and S_2 of a face dataset (filtered for duplicates). The subset size N varies from 1 to 10^5. We then generate a sample from each model with a reverse diffusion algorithm (SISA), initialized from the same noise image. **Top.** For training sets of size $N = 1$ to $N = 100$, the networks memorize, producing samples nearly identical to examples from the training set. For $N = 1000$, generated samples are similar to a training example, but show distortions in some regions. This transitional regime corresponds to a qualitative change in the shape of the PSNR curve (Figure 3.2). For $N = 10^5$, the two networks generate nearly identical samples, which no longer resemble images in their corresponding training sets. **Bottom.** The distribution of cosine similarity (normalized inner product) between pairs of images generated by the two networks (blue) shifts from left to right with increasing N, showing vanishing model variance. Conversely, the distribution of cosine similarity between generated samples and the most similar image in their corresponding training set (orange) shifts from right to left.

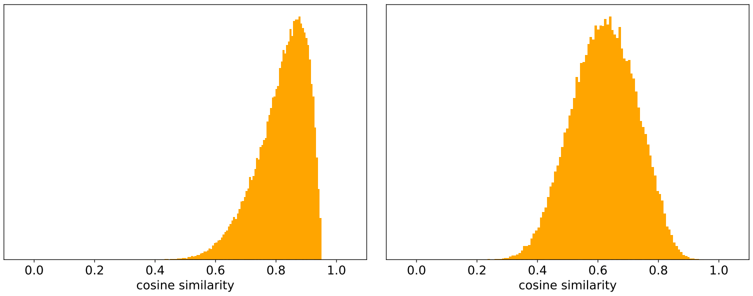


Figure 3.4: Similarity between data subsets. Histogram of cosine similarity between pairs of closest images in the non-overlapping subsets S_1 and S_2 of CelebA (left) and LSUN bedroom (right). Images with similarity score higher than 0.95 are removed from the datasets before training to eliminate replicated images. This should be compared with the histograms in Figures 3.3 and 3.7.

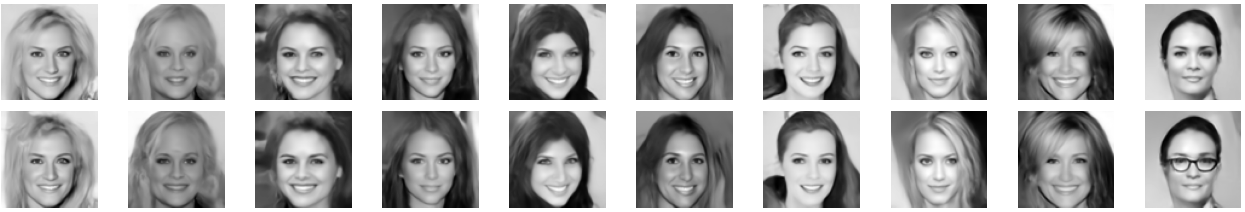


Figure 3.5: More examples to illustrate convergence of model variance for models shown in Figure 3.3, at $N = 10^5$. Samples generated by each denoiser are shown in separate rows, where each column shows same initialization across the networks. The networks generate nearly identical samples, showing convergence to the same function.

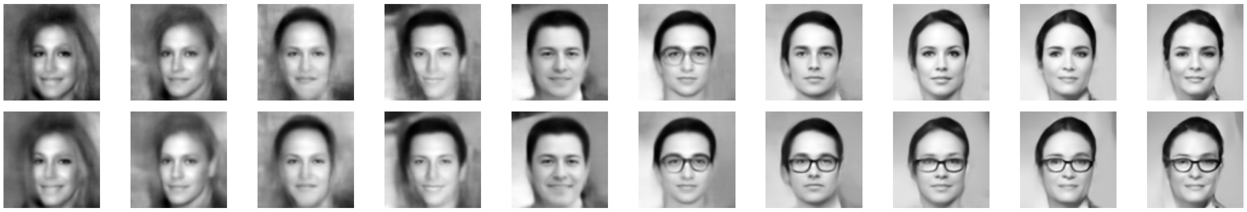


Figure 3.6: Bifurcation of trajectories. Sampling trajectories for the two samples shown in the last column of Figure 3.5. The two diffusion models arrive at different samples starting from the same initial point. The bifurcation of gradients appears to emerge somewhere around the middle of the trajectories, which illustrates instabilities predicted by recent dynamical models (Biroli et al., 2024). All the intermediate samples in the trajectories have been denoised in a on-shot denoising manner using the corresponding denoisers. This example shows that the convergence is not perfect, hence the distribution of cosine similarities at $N = 10^5$ is not perfectly a delta function at 1.

The minimum size of the training set, N, for which the model transitions from memorization to generalization indeed depends on the architecture, image size and complexity of data distribution. Nevertheless, with enough data, two models trained on non-overlapping subsets of data converge to virtually the same function. Deriving precise scaling laws for number of images required for generalization as a function of network capacity, dataset complexity, and image dimensionality is an interesting and important topic for future investigation. Figure 3.12, shows a preliminary result in scaling law of UNet architecture on CelebA dataset.

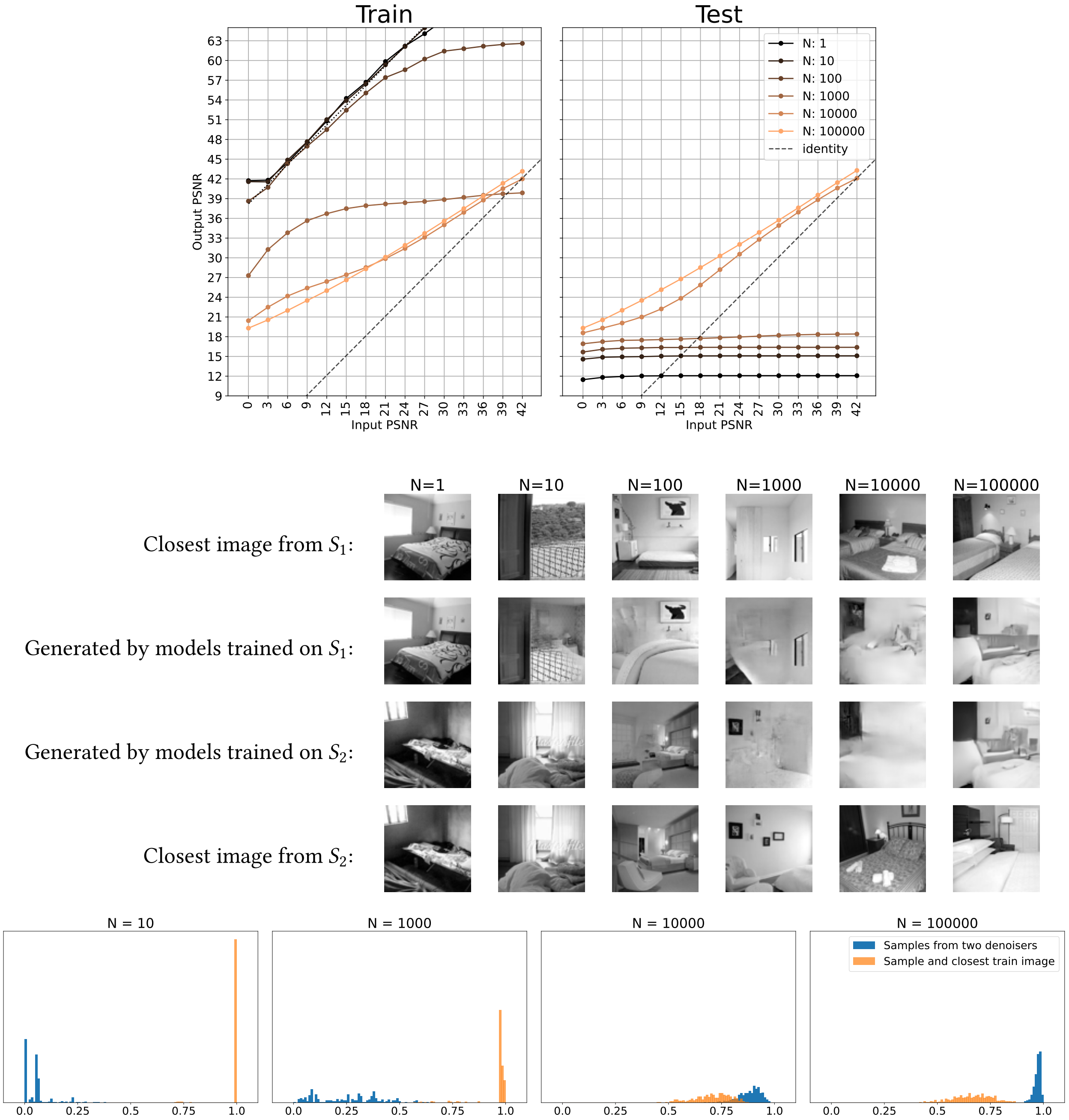


Figure 3.7: Transition from memorization to generalization, for a UNet denoiser trained on bedroom LSUN images (Yu et al., 2015) downsampled to 80 × 80. Similarly to denoisers trained on face images shown in Figure 3.2, the model transitions from memorizing the training set to generalizing outside of the training set. **Top.** At $N = 10^5$ the performance is almost identical on training and test sets, and the model is no longer overfitting the training data. **Middle.** Diffusion models are trained on non-overlapping subsets S_1 and S_2 of a bedroom LSUN dataset. The subset size N varies from 1 to 10^5. Notice the samples generated by network trained on $N = 100$ images: they are combinations of patches of training images. This type of memorization has been previously reported in (Somepalli et al., 2023). See caption of Figure 3.3 for a complete description of the figure.

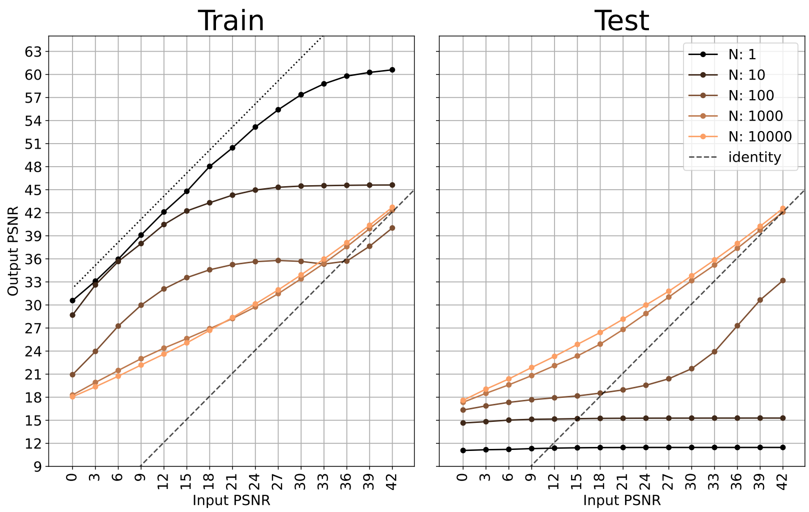


Figure 3.8: Transition from memorization to generalization, for a BF-CNN denoiser trained on CelebA HQ dataset (Karras et al., 2018) downsampled to 40 × 40 resolution. See caption of Figure 3.2.

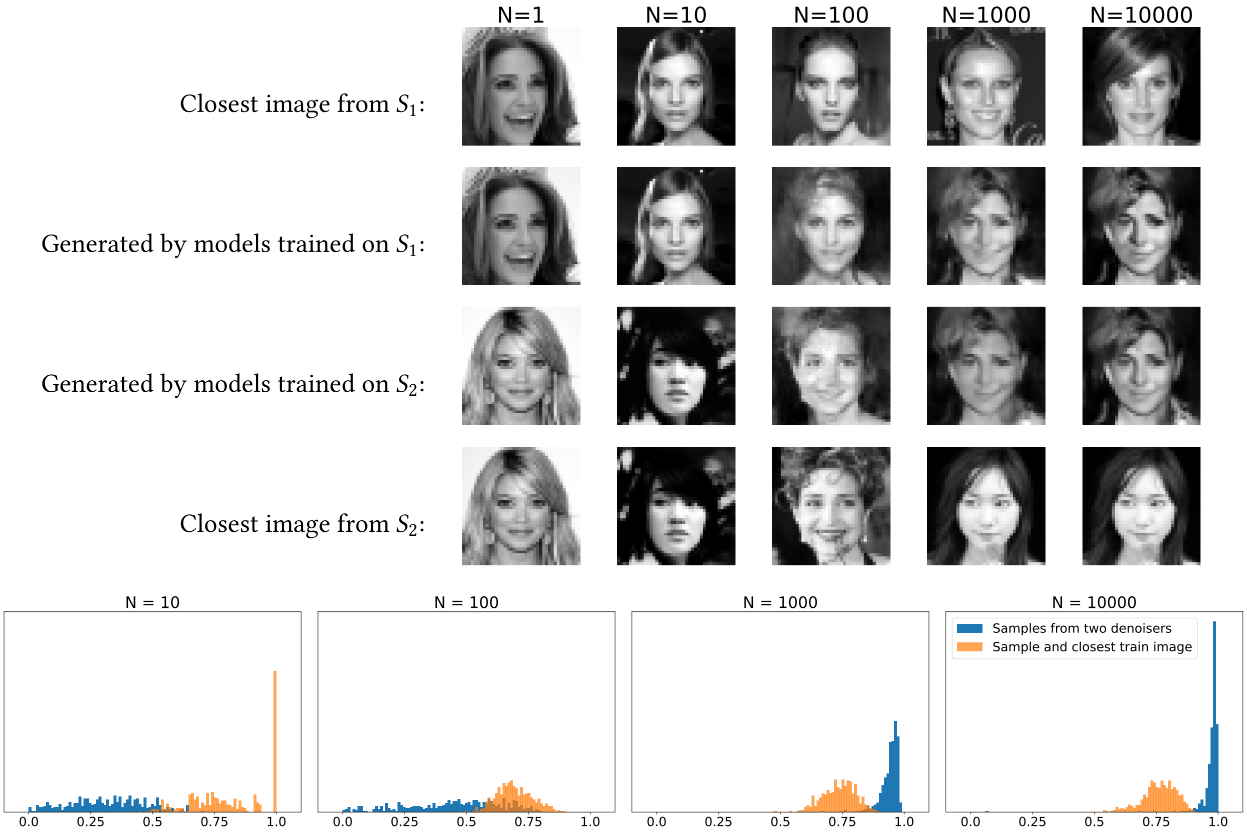


Figure 3.9: Convergence of model variance for a BF-CNN denoiser. BF-CNN denoisers are trained on non-overlapping subsets S_1 and S_2 of CelebA HQ dataset. The subset size N varies from 1 to 10^4. See caption of Figure 3.3.

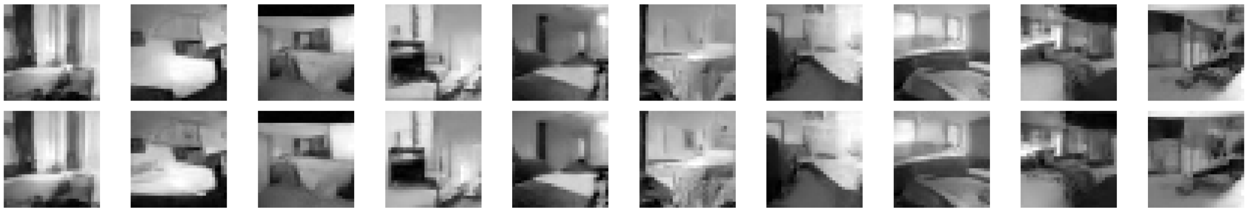


Figure 3.10: Convergence of model variance on LSUN bedroom dataset (Yu et al., 2015). A dataset of bedroom images is partitioned into two non-overlapping datasets, S_1 and S_2, each containing $N = 20,000$ images down-sampled to size 32×32. We train two networks (BF-CNN architecture described in Appendix B.1) on S_1 and S_2. Each network is then used in an iterative deterministic reverse diffusion algorithm to generate a sample, with both networks initialized with the same noise image. Samples generated by each denoiser are shown in separate rows, where each column shows same initialization across the networks. The networks generate nearly identical samples, showing convergence to the same function.

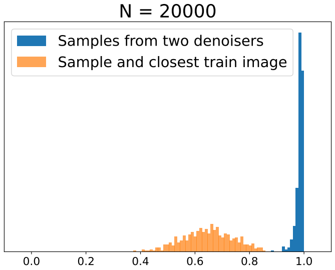


Figure 3.11: Convergence for LSUN dataset. Blue histogram: cosine similarity between samples generated by two denoisers trained on non-overlapping training sets of size $N = 20,000$ from LSUN bedroom dataset downsampled to 32×32 resolution. Orange histograms: cosine similarity between generated samples and the closest image from the corresponding training set. Images drawn from the two denoisers are very similar to each other, compared to the closest image in their respective training sets.

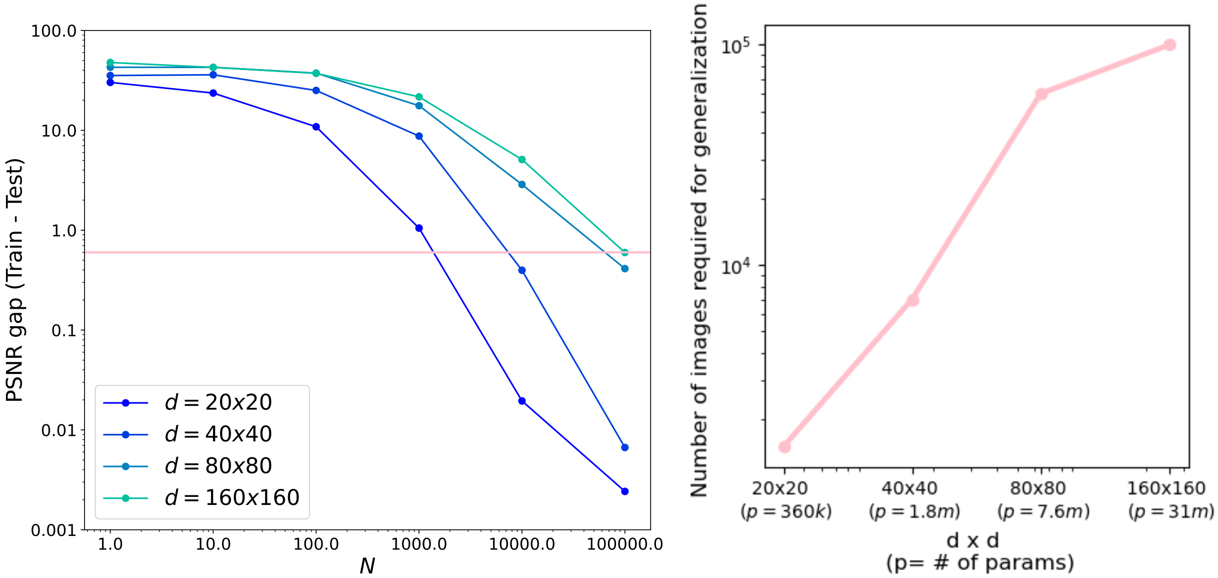


Image resolution	number of encoder decoder blocks	Receptive field size	number of parameters
20×20	1	18×18	$360k$
40×40	2	44×44	$1.8m$
80×80	3	92×92	$7.6m$
160×160	4	188×188	$31m$

Figure 3.12: Generalization scaling law. Generalization is measured as the difference between train and test PSNRs averaged across noise levels $\sigma \in [0, 1]$. **Top left.** Each curve shows average PSNR gap as a function training set size for a specific image resolution $d \times d$ and number of network parameters (which grow together by a factor of 4). As expected, to reach the threshold PSNR gap, denoisers trained on larger images require more training data. **Top right.** Number of training images required to hit the PSNR gap threshold as a function of image size and number of network parameters, which grow together by a factor of 4. Number of images grows linearly on a log-log plot. However, it seems that for larger images the relationship becomes sublinear. This observation is consistent with results in Chapter 4 indicating that conditioned on coarser content of the image, learning the finer details requires less data due to the conditional Markov property of images. **Bottom.** UNet architectures used in experiments shown here. With the four-fold increase of the image size, the number of parameters increases approximately four times.

3.2 Inductive biases

The number of samples needed for estimation of an arbitrary probability density grows exponentially with dimensionality (the “curse of dimensionality”). As a result, estimating high-dimensional distributions is only feasible if one imposes strong constraints or priors over the hypothesis space. In a diffusion model, these arise from the network architecture and the optimization algorithm, and are referred to as the inductive biases of the network (Goyal and Bengio, 2022; Griffiths et al., 2023; Wilson and Izmailov, 2020). In Section 3.1.2, we demonstrated that DNN denoisers can learn scores (and thus a density) from relatively small training sets. This generalization result, combined with the high quality of sampled images, is evidence that the inductive biases are well-matched to the “true” distribution of images, allowing the model to rapidly converge to a good solution through learning. On the contrary, when inductive biases are not aligned with the true distribution, the model will arrive at a poor solution with high model bias.

For diffusion methods, learning the right density model is equivalent to performing optimal denoising at all noise levels (see Section 3.1.1). The inductive biases on the density model thus arise directly from inductive biases in the denoiser. This connection offers a means of evaluating the accuracy of the learned probability models, which is generally difficult in high-dimensions.

3.2.1 Denoising as shrinkage in an adaptive basis

The inductive biases of the DNN denoiser can be studied through an eigendecomposition of its Jacobian. We describe the general properties that are expected for an optimal denoiser, and examine several specific cases for which the optimal solution is partially known.

Jacobian eigenvectors as an adaptive basis. To analyze inductive biases, we perform a local analysis of a denoising estimator $\hat{x}(y) = f(y)$ by looking at its Jacobian $\nabla f(y)$. For simplicity, we assume that the Jacobian is symmetric and non-negative (we show below that this holds for the optimal denoiser, and it is approximately true of the network Jacobian (Mohan* et al., 2020)). We can then diagonalize it to obtain eigenvalues $(\lambda_k(y))_{1\leq k\leq d}$ and eigenvectors $(e_k(y))_{1\leq k\leq d}$.

In Chapter 2 we saw if $f(y)$ is computed with a DNN denoiser with no additive "bias" parameters, its input-output mapping is piecewise linear, as opposed to piecewise affine (Mohan* et al., 2020; Romano et al., 2017a). It follows that the denoiser mapping can be rewritten in terms of the Jacobian eigendecomposition as

$$f(y) = \nabla f(y)\, y = \sum_k \lambda_k(y)\ \langle y, e_k(y)\rangle\ e_k(y). \tag{3.4}$$

The denoiser can thus be interpreted as performing shrinkage with factors $\lambda_k(y)$ along axes of a basis specified by $e_k(y)$. Note that both the eigenvalues and eigenvectors depend on the noisy image y (i.e., both the basis and shrinkage factors are *adaptive* (Milanfar, 2012b)).

Even if the denoiser is not bias-free, small eigenvalues $\lambda_k(y)$ reveal local invariances of the denoising function: small perturbations in the noisy input along the corresponding eigenvectors $e_k(y)$ do not affect the denoised output. Intuitively, such invariances are a desirable property for a denoiser, and they are naturally enforced by minimizing mean squared error (MSE) as expressed with Stein's unbiased risk estimate (SURE, proved in Appendix B.3.3 for completeness):

$$\mathrm{MSE}(f, \sigma^2) = \mathop{\mathbb{E}}_{y}\left[2\sigma^2 \operatorname{tr} \nabla f(y) + \|y - f(y)\|^2 - \sigma^2 d\right]. \tag{3.5}$$

To minimize MSE, the denoiser must trade off the approximate "rank" of the Jacobian (the trace is the sum of the eigenvalues) against an estimate of the denoising error: $\|y - f(y)\|^2 - \sigma^2 d$. The denoiser thus locally behaves as a (soft) projection on a subspace whose dimensionality corresponds to the rank of the Jacobian. As we now explain, this subspace approximates the support of the posterior distribution $p(x|y)$, and thus gives a local approximation of the support of $p(x)$.

It is shown in Appendix B.3.1 that the optimal minimum MSE denoiser and its Jacobian are given by

$$f^\star(y) = y + \sigma^2 \nabla \log p_\sigma(y) = \mathop{\mathbb{E}}_{x}[x|y], \tag{3.6}$$

$$\nabla f^\star(y) = \mathrm{Id} + \sigma^2 \nabla^2 \log p_\sigma(y) = \sigma^{-2} \mathrm{Cov}\,[x \mid y]\,. \tag{3.7}$$

That is, the Jacobian of the optimal denoiser is proportional to the posterior covariance matrix, which is symmetric and non-negative. This gives us another interpretation of the adaptive eigenvector basis as providing an optimal approximation of the unknown clean image x given the noisy observation y. Further, the optimal denoising error is then given by (see Appendix B.3.1 for the first equality)

$$\mathrm{MSE}(f^\star, \sigma^2) = \mathop{\mathbb{E}}_{y}\left[\mathrm{tr}\,\mathrm{Cov}\,[x|y]\right] = \sigma^2 \mathop{\mathbb{E}}_{y}\left[\mathrm{tr}\,\nabla f^\star(y)\right] = \sigma^2 \mathop{\mathbb{E}}_{y}\left[\sum_k \lambda_k^\star(y)\right]. \tag{3.8}$$

A small denoising error thus implies an approximately low-rank Jacobian (with many small eigenvalues) and thus an efficient approximation of x given y.

In most cases, the optimal adaptive basis $(e_k^\star(y))_{1\le k\le d}$ is not known. Rather than aiming for exact optimality, classical analyses (Donoho, 1995) thus focus on the asymptotic decay of the denoising error as the noise level σ^2 falls, up to multiplicative constants. This corresponds to finding a basis $(e_k(y))_{1\le k\le d}$ which captures the asymptotic slope of the PSNR plots in Figure 3.2 but not necessarily the intercept. This weaker notion of optimality is obtained by showing matching upper and lower-bounds on the asymptotic behavior of the denoising error. To provide intuition, we first consider a fixed orthonormal basis $e_k(y) = e_k$, and then consider the more general case of best bases selected from a fixed dictionary.

Denoising in a fixed basis. Consider a denoising algorithm that is restricted to operate in a fixed basis e_k but can adapt its shrinkage factors $\lambda_k(y)$. An unreachable lower-bound on the denoising error—and thus an upper-bound on the PSNR slope—is obtained by evaluating the performance of an "oracle" denoiser where the shrinkage factors λ_k depend on the unknown clean image x rather than the noisy observation y (Mallat, 2008). Appendix B.3.4 shows that the denoising error of this oracle is

$$\mathop{\mathbb{E}}_{x}\left[\sum_k \left((1-\lambda_k(x))^2 \langle x, e_k\rangle^2 + \lambda_k(x)^2\sigma^2\right)\right], \tag{3.9}$$

which is minimized when $\lambda_k(x) = \frac{\langle x,e_k\rangle^2}{\langle x,e_k\rangle^2+\sigma^2}$. The coefficient $\lambda_k(x)$ thus acts as a soft threshold: $\lambda_k(x) \approx 1$ when the signal dominates the noise and $\lambda_k(x) \approx 0$ when the signal is weaker than the noise. Appendix B.3.4 then shows that the oracle denoising error is the expected value of

$$\sigma^2 \sum_k \lambda_k(x) = \sum_k \frac{\sigma^2 \langle x, e_k\rangle^2}{\langle x, e_k\rangle^2 + \sigma^2} \sim \sum_k \min(\langle x, e_k\rangle^2, \sigma^2) = M\sigma^2 + \|x - x_M\|^2, \tag{3.10}$$

where $x_M = \sum_{\langle x,e_k\rangle^2>\sigma^2} \langle x, e_k\rangle\, e_k$ is the M-term approximation of x with the M basis coefficients $\langle x, e_k\rangle$ above the noise level, and $\sim$ means that the two terms are of the same order up to multiplicative constants (here smaller than 2). The denoising error is small if x has a sparse representation in the basis, so that both M and the approximation error $\|x - x_M\|^2$ are small. For example, if the coefficients decay as $\langle x, e_k\rangle^2 \sim k^{-(\alpha+1)}$ (up to reordering), Appendix B.3.4 shows that

$$M\sigma^2 + \|x - x_M\|^2 \sim \sigma^{2\alpha/(\alpha+1)}, \tag{3.11}$$

which is a lower bound on the MSE of any denoising algorithm in the basis e_k. Reciprocally, this oracle denoising error is nearly reached with a soft-thresholding estimator that computes the shrinkage factors $\lambda_k(y)$ by comparing $\langle y, e_k\rangle^2$ (rather than $\langle x, e_k\rangle^2$) with a threshold proportional to σ^2 (Donoho and Johnstone, 1994), and achieves the decay (3.11) up to a logarithmic factor. The decay (3.11) of the MSE with decreasing σ corresponds to an asymptotic slope of $\alpha/(\alpha+1)$ in the PSNR curve when the input PSNR increases. Thus, a larger sparsity/regularity exponent α, which corresponds to a faster decay of the small coefficients of x in the basis $(e_k)_{1\leq k\leq d}$, leads to improved denoising performance.

Best adaptive bases. Adapting the basis $(e_k)_{1\leq k\leq d}$ to the noisy image y allows obtaining sparser representations of the unknown clean image x with a faster decay, and thus a larger PSNR slope. To calculate the optimal adaptive basis, we need to find an oracle denoiser that has the same asymptotic MSE as a non-oracle denoiser, yielding matching lower and upper bounds on the asymptotic MSE.

Consider an oracle denoiser which performs a thresholding in an oracle basis $(e_k(x))$ that depends on the unknown clean image x. The above analysis then still applies, and if the coefficients $\langle x, e_k(x)\rangle^2$ decay as $k^{-(\alpha+1)}$, then the asymptotic PSNR slope is again $\alpha/(\alpha+1)$. The best oracle basis satisfies $e_1(x) = x/\|x\|$, but it yields a loose lower bound as it cannot be estimated from the noisy image y alone. We thus restrict the oracle denoiser to choose the basis $(e_k(x))$ within a fixed dictionary. A larger dictionary increases adaptivity, but it then becomes harder to estimate the basis that best represents x from y alone. If the dictionary of bases is constructed from a number of vectors e_k which is polynomial in the dimension d (the number of bases can however be exponential in d) then a thresholding in the basis $(e_k(y))$ that best approximates the noisy image y achieves the same slope as the oracle denoiser (Barron et al., 1999; Dossal et al., 2011). This near-optimality despite the presence of noise comes from the limited choice of possible basis vectors e_k in the dictionary, which limits the variance of the best-basis estimation, e.g. by preventing $e_1(y) = y/\|y\|$. The main difficulty is then to design a small-enough dictionary that gives optimal representations of images from the data distribution in order to achieve the optimal PSNR slope.

We now evaluate the inductive biases of DNN denoisers through this lens. In Section 3.1, we

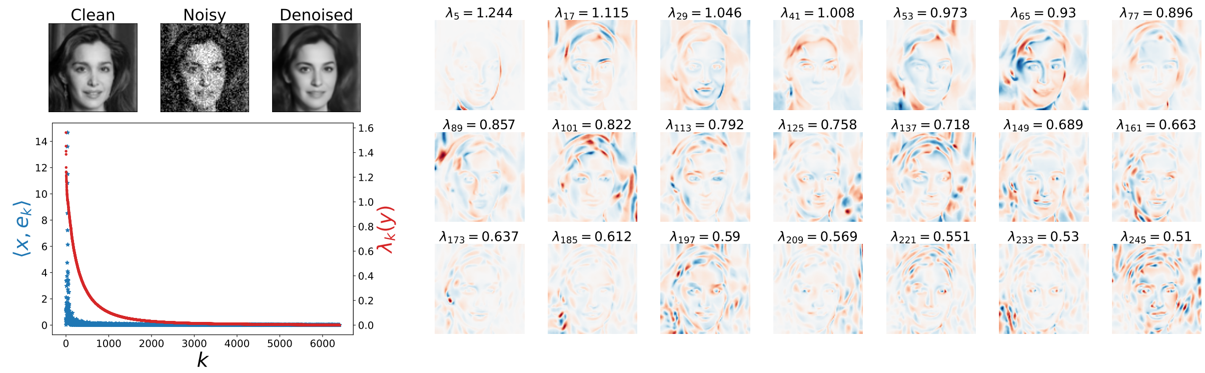


Figure 3.13: Analysis of a denoiser trained on 10^5 face images, evaluated on a noisy test image. **Top left.** Clean, noisy ($\sigma = 0.15$) and denoised images. **Bottom left.** Decay of shrinkage values $\lambda_k(y)$ (red), and corresponding coefficients $\langle x, e_k(y) \rangle$ (blue), evaluated for the noisy image y. The rapid decay of the coefficients indicates that the image content is highly concentrated within the preserved subspace. **Right.** The adaptive basis vectors $e_k(y)$ contain oscillating patterns, adapted to lie along the contours and within smooth regions of the image, whose frequency increases as $\lambda_k(y)$ decreases.

showed that the DNN denoisers overcome the curse of dimensionality: their variance decays to zero in the generalization regime. In the next section, we explain this observation by demonstrating that they are inductively biased towards adaptive bases $e_k(y)$ from a particular class.

3.2.2 Geometry-adaptive harmonic bases in DNNs

Figure 3.13 shows the shrinkage factors ($\lambda_k(y)$), adaptive basis vectors ($e_k(y)$), and signal coefficients ($\langle x, e_k(y) \rangle$) of a DNN denoiser trained on 10^5 face images. The eigenvectors have oscillating patterns both along the contours and in uniformly regular regions and thus adapt to the geometry of the input image. We call this a geometry-adaptive harmonic basis (GAHB). The coefficients are sparse in this basis, and the fast rate of decay of eigenvalues exploits this sparsity. The high quality of generated images and the strong generalization results of Section 3.1 show that DNN denoisers rely on inductive biases that are well-aligned to photographic image distributions. All of this suggests that DNN denoisers might be inductively biased towards GAHBs. In the following, we provide evidence supporting this conjecture by analyzing networks trained on synthetic datasets where the optimal solution is (approximately) known.

$\mathbf{C}^\alpha$ images and bandlet bases. If DNNs are inductively biased towards GAHBs, we expect that they generalize and converge to the optimal denoising performance when such bases are optimal. We consider the so-called geometric $\mathbf{C}^\alpha$ class of images (Donoho, 1999; Korostelev and Tsybakov, 1993; Peyré and Mallat, 2008) which consist of regular contours on regular backgrounds, where the degree of regularity is controlled by α. Examples of these images are shown in Figure 3.14 and Appendix B.2.1. A mathematical definition and an algorithm for their synthesis are presented in Appendix B.4.

Optimal sparse representations of $\mathbf{C}^\alpha$ images are obtained with "bandlet" bases (Peyré and Mallat, 2008). Bandlets are harmonic functions oscillating at different frequencies, whose geometry is adapted to the directional regularity of images along contours. Geometric $\mathbf{C}^\alpha$ images can be represented with few bandlets having low-frequency oscillations in regular regions and along contours but sharp variations across contours. The k-th coefficient in the best bandlet basis then decays as $k^{-(\alpha+1)}$. It follows that the optimal denoiser has a PSNR which asymptotically increases with a slope $\alpha/(\alpha+1)$ as a function of input PSNR (Dossal et al., 2011; Korostelev and Tsybakov, 1993).

Figure 3.14 shows that DNN denoisers trained on $\mathbf{C}^\alpha$ images also achieve this optimal rate and learns GAHBs, similarly to bandlets but with a more flexible geometry. This generalization performance confirms that inductive biases of DNNs favor GAHBs.

Low-dimensional manifolds. If DNNs are inductively biased towards GAHBs, then we expect these bases to emerge even in cases where they are suboptimal. To test this prediction, we consider a dataset of disk images with varying positions, sizes, and foreground/background intensities. This defines a five-dimensional *curved* manifold, with a tangent space evaluated at a disk image x that is spanned by deformations of x along these five dimensions. When the noise level σ is much smaller than the radius of curvature of the manifold, the posterior distribution $p(x|y)$ is supported on an approximately flat region of the manifold, and the optimal denoiser is

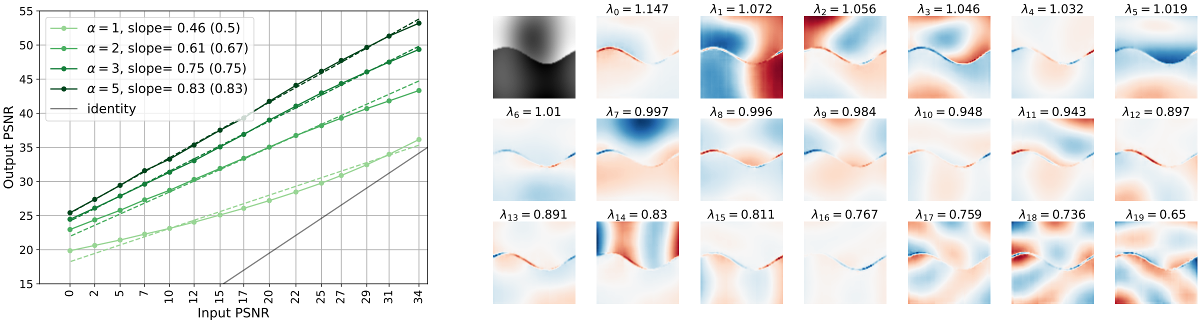


Figure 3.14: Analysis of denoiser trained on C^α images. UNet denoisers trained on 10^5 C^α images achieve near-optimal performance. **Left.** PSNR curves for various regularity levels α. The empirical slopes closely match the theoretical optimal slopes (parenthesized values, dashed lines). **Right.** A C^α image ($\alpha = 4$) of size 80×80 and its top eigenvectors, which consist of harmonics on the two regions and harmonics along the boundary. The frequency of the harmonics increases with k. More examples are given in Appendix B.2.1.

approximately a projection onto the tangent space. Thus, the optimal Jacobian should have only five non-negligible eigenvalues, whose corresponding eigenvectors span the tangent space. The remaining eigenvectors should have shrinkage factors of $\lambda = 0$, but are otherwise unconstrained. The optimal MSE is asymptotically equal to $5\sigma^2$, corresponding to a PSNR slope of one.

Figure 3.17 shows an analysis of a denoiser trained on 10^5 disk images, of size 80×80. We observe additional basis vectors with non-negligible eigenvalues that have a GAHB structure, with oscillations on the background region and along the contour of the disk. We also find that the number of non-zero eigenvalues *increases* as the noise level decreases, leading to a suboptimal PSNR slope that is less than 1.0. These results reveal that the inductive biases of the DNN are not perfectly aligned with low-dimensional manifolds, and that in the presence of the curvature, this suboptimality increases as the noise level decreases.

We obtain similar results on two additional examples of a distribution supported on a low-dimensional manifold, given in Figure 3.15 and Figure 3.16.

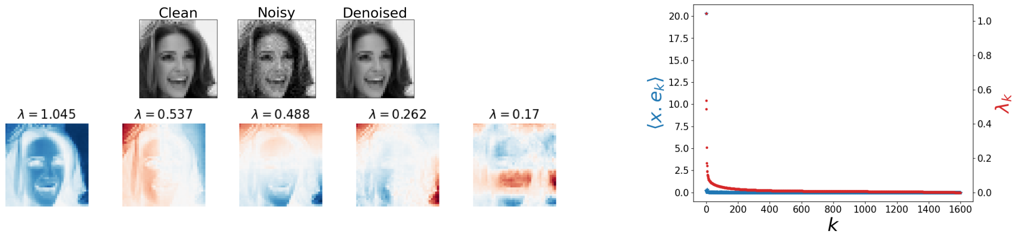


Figure 3.15: Analysis of a BF-CNN denoiser trained on a single face image. BF-CNN denoiser trained on a single face image, with intensity rescaling. We consider an image class consisting of a single image $x \in \mathbb{R}^d$ and its positive rescalings $s\,x$ for $s > 0$. The resulting images lie on a ray emanating from the origin, and optimal denoising corresponds to projecting the noisy image onto this ray. The optimal denoising basis should therefore include the normalized vector $x/\|x\|$ with associated shrinkage factor $\lambda = 1$, whereas the remaining basis vectors should have shrinkage factors of $\lambda = 0$ but are otherwise unconstrained. This optimal denoiser achieves an MSE of σ^2, and thus a linear PSNR curve with unit slope and intercept $10\log_{10}(d)$. **Top left**. Denoising of the training image with $\sigma = 0.04$. **Right**. Decay of the coefficients $\langle x, e_k \rangle$ and the shrinkage factors λ_k. The DNN denoiser exhibits a slower decay of shrinkage factors than the optimal solution, which results in suboptimal performance. **Bottom left**. Top 5 basis vectors $e_k(y)$. The first basis vector is nearly identical to the (normalized) train image. The next vectors, which have non-zero shrinkage factors, exhibit 2D harmonics. These GAHB components underlie the non-optimal behavior of the denoiser. Specifically, the $N = 1$ curve in the left panel of Figure 3.8 shows that performance as a function of noise level falls below the optimal solution (dotted line). The DNN performance has a unit slope over most of the noise range but has a less-than-optimal intercept (the flattening of the curve at small noise levels is a result of de-emphasis of small noise levels during training).

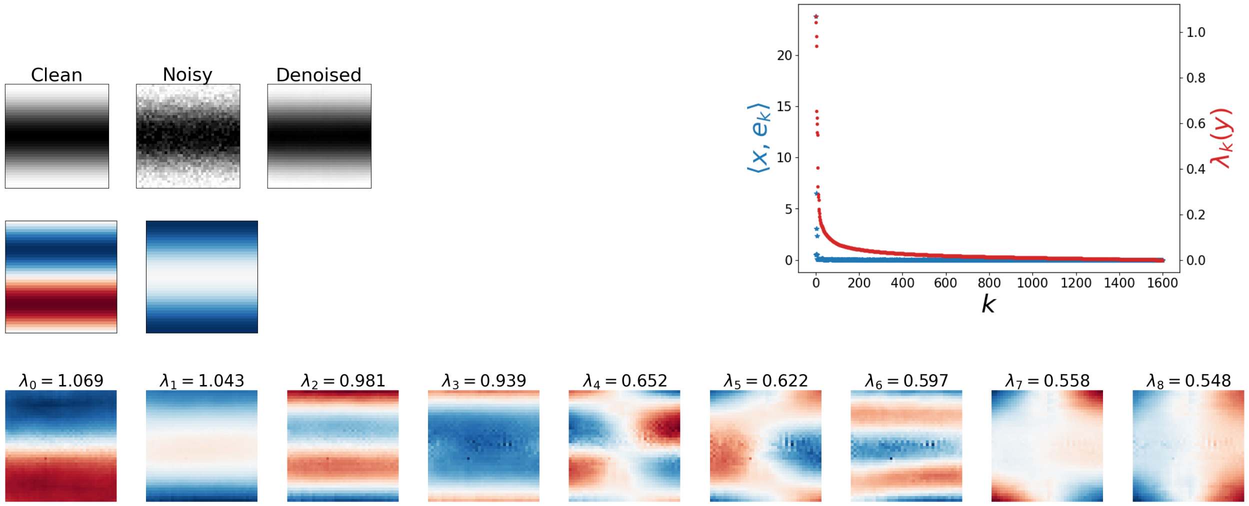


Figure 3.16: Analysis of a BF-CNN denoiser is trained on a set of 2D sine wave images. A BF-CNN denoiser is trained on a set of 2D sine wave images with unit frequency and varying phases and intensities. The train images thus lie on a 2D cone manifold with low curvature. For small σ, the manifold can be assumed to be locally flat, so that the optimal denoising is achieved by projecting the noisy image on the two-dimensional subspace tangent to the manifold. This subspace is spanned by two sine waves with unit frequency and a $\pi/2$ phase shift. **Top left.** Clean, noisy ($\sigma = 0.08$), and denoised test image. **Middle left.** The unit vectors spanning the tangent subspace. The optimal denoising results from projection onto this subspace. **Bottom row.** Empirical basis obtained from the network Jacobian. The empirical solution has a slower decay than optimal (i.e., $\langle x, e_k(y) \rangle > 0$ for $k \geq 2$, as seen in the **right** panel), with harmonic patterns. This sub-optimality reveals the nature of the inductive bias.

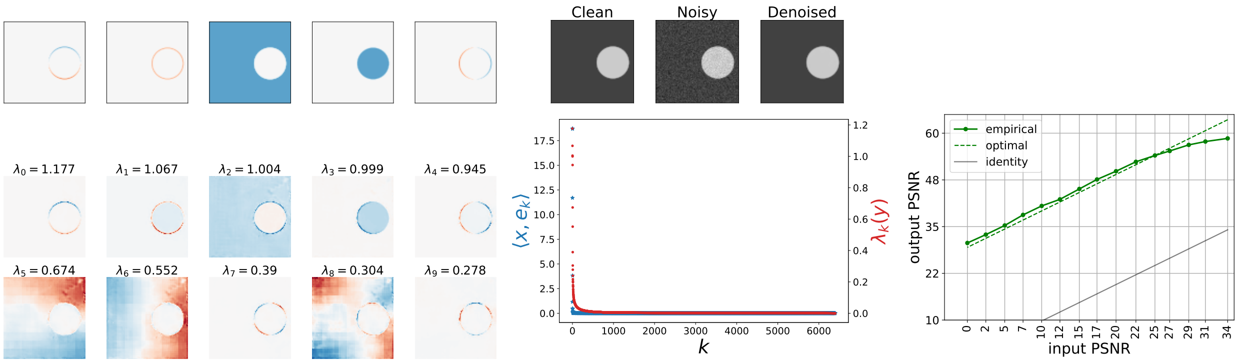


Figure 3.17: Analysis of a denoiser trained on a disk dataset. UNet denoiser trained on a dataset of translating and dilating disks, with variable foreground/background intensity. **Top center.** Clean, noisy ($\sigma = 0.04$), and denoised images. **Bottom center.** The decay of shrinkage factors $\lambda_k(y)$ and coefficients $\langle x, e_k(y) \rangle$ indicates that the network achieves and preserves a sparse representation of the true image. **Top right.** denoising performance is sub-optimal, with PSNR slope below the optimal value of 1.0 for small noise. **Top left.** An optimal basis (in the small-noise limit) spanning the 5-dimensional tangent space of the image manifold. **Bottom left.** Top eigenvectors of the adaptive basis. The first five basis vectors closely match the basis of the tangent space of the manifold evaluated at the clean image. In contrast, the next five are GAHBs that lie along contours and within background regions of the clean image.

Shuffled faces. We also consider a dataset of shuffled faces, obtained by applying a common permutation to the pixels of each face image. This permutation does not preserve locality between neighboring pixels, and thus the optimal basis does not have harmonic structure. The resulting mismatch between the DNN inductive biases and the data distribution result in substantially worse performance than for the original (unscrambled) faces.

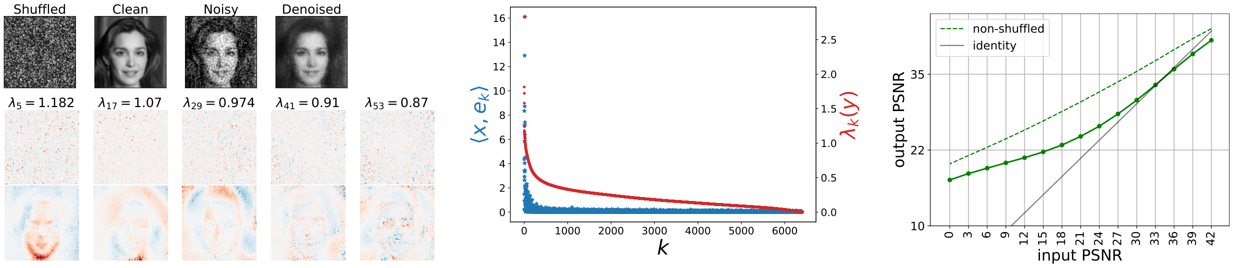


Figure 3.18: Analysis of a denoiser trained on a dataset of shuffled faces. DNN denoiser trained on a dataset of shuffled faces obtained by permuting the pixels of 10^5 face images in the CelebA dataset. The permutation was chosen randomly, and does not preserve locality, as neighboring pixels are mapped to independent positions. By construction, the optimal denoiser on shuffled faces has the same performance as the optimal denoiser on ordinary faces (unshuffling the image pixels, optimally denoising the face image, and then shuffling the pixels back). For visualization purposes, we “unshuffle” the pixels by applying the inverse of the permutation to the images before display. **Top left.** Clean (shuffled then unshuffled), noisy (unshuffled, $\sigma = 0.3$), and denoised (unshuffled) images. **Middle.** The shrinkage factors $\lambda_k(y)$ decay more slowly than when the denoiser is trained on non-shuffled faces (Figure 3.13), which is indicative of suboptimality.. **Right.** The denoiser performs significantly worse than the denoiser trained on unshuffled faces: the MSE is much higher with a much lower PSNR slope. **Bottom left.** Basis vectors (top row: shuffled, bottom row: unshuffled). After unshuffling, we observe GAHBs adapted to the geometry of the face, although these are noisier and less precisely aligned with the image features than the non-shuffled examples in Figure 3.13.

3.3 Discussion

Diffusion generative models, which operate through iterative application of a trained DNN denoiser, have recently surpassed all previous methods of learning probability models of images. Their training objective (minimization of squared denoising error) is simple and robust, and they generate samples of impressive quality. In this paper, we elucidate the approximation properties that underlie this success, by analyzing the trained denoiser, which is directly related to the score function, and to the density from which the samples are drawn.

We show empirically that diffusion models memorize samples when trained on small sets, but transition to a strong form of generalization as the training set size increases, converging to a unique density model that is independent of the specific training samples. The amount of data needed to reach this phase transition is very small relative to the size of dataset needed for convergence without any inductive biases, and depends on the image size and complexity relative to the neural network capacity (Yoon et al., 2023). It is of interest to extend both the theory and the empirical studies to account for the interplay of these factors. Figure 3.12 shows preliminary results in this direction.

We also examined the inductive biases that enable this strong generalization. Using a well-established mathematical framework, we showed that DNN denoisers perform shrinkage of noisy coefficients in a geometry-adaptive harmonic basis (GAHB) which is shaped by geometric features of the image. For the $\mathbf{C}^{\alpha}$ class of images, such geometric bases are known to be optimal, and DNN denoisers achieve near-optimal performance on this class. Previous mathematical literature has shown that bandlet bases, which are a specific type of GAHB, are near-optimal for this class, but the GAHBs learned by the DNN denoiser are more general and more flexible. For images drawn from low-dimensional manifolds, for which the optimal basis spans the tangent subspace of the manifold, we find that DNN denoisers achieve good denoising within a basis aligned with this subspace, but also incorporate GAHB vectors in the remaining unconstrained dimensions. The non-suppressed noise along these additional GAHB components leads to suboptimal denoising performance. This observation, along with similar ones shown in Figures 3.15 and 3.16, provide more supporting evidence for the hypothesis that inductive biases of DNN denoisers promote GAHBs.

We do not provide a formal mathematical definition of the class of GAHBs arising from the inductive biases of DNNs. Convolutions in DNN architectures, whose eigenvectors are sinusoids, presumably engender GAHB harmonic structure, but the geometric adaptivity must arise from interactions with rectification nonlinearities (ReLUs). A more precise elucidation of this GAHB function class, and its role in shaping inductive biases of the DNNs used in a wide variety of other tasks and modalities, is of fundamental interest.

4 Muli-scale Local conditional image density

In the previous two chapters, we demonstrated that a DNN denoiser embeds a density which is accessible through sampling, and then showed, when in generalization regime, the embedded density is a good approximation of the underlying image prior. In this chapter, we employ this learned image density to study the question of low-dimensionality of image priors.

A critical feature for the success of DNNs in the diffusion framework is their global receptive fields (RF). Large RF allows for capturing long range dependencies in the image through modeling the joint density directly. Naturally, this yields to a large number of parameters, currently on the order of hundreds of millions for the state-of-the-art diffusion models. Can we exploit image properties to factorize the joint density into low dimensional densities, thereby reducing the number of parameters and training examples?

To answer this question, we return to the classical idea of modeling image densities using Markov random fields (MRF) (Dobrushin, 1968; Sherrington and Kirkpatrick, 1975). Markov random fields assume localized conditional dependencies, which guarantees that the density can be factorized into terms acting on local, typically overlapping neighborhoods (Clifford and

A version of this chapter is published (Kadkhodaie et al., 2023a) in collaboration with Florentin Guth and Stephane Mallat. The code for generating results can be found at https://github.com/LabForComputationalVision/local-probability-models-of-images

Hammersley, 1971). MRFs 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 $\boldsymbol{x}(u)$, conditioned on pixel values $\boldsymbol{x}(v)$ 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., (Cui and Wang, 2005; Geman and Geman, 1984; Malfait and Roose, 1997)), but none have demonstrated performance comparable to recent deep networks while retaining a local dependency structure.

In this chapter, first we delve into a historical account of MRFs in image processing. The goal is to show that the concepts on which MRFs rest—stationarity, locality and multi-scale conditioning—are still of fundamental importance, despite MRF's inferior performance. Then, we combine these concepts with the power of deep nets to constrain and study score-based diffusion models.

To this end, we develop a low-dimensional probability model for images decomposed into multi-scale wavelet sub-bands. 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 effective size of Markov neighborhoods (i.e. the size w.r.t to the grid size) grows from fine to coarser scales. 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 our coarse-to-fine sampling strategy for drawing samples from the posterior, 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 for images can be reduced to low-dimensional Markov conditional models, alleviating the curse of dimensionality.

4.1 Theory of Markrov random fields

Markov random fields are undirected graphical models that satisfy the Markov property. Let $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a graph, where $\mathcal{S} = \{s_1, s_2, ...s_N\}$ are a set of vertices or sites and $\mathcal{E}$ is the set of edges. Markov property indicates that at any site s on the graph, the value $x_s \in \Lambda$ of the random variable, X_s, conditioned on its neighbors, is independent from all variables outside its neighborhood. Below, we make these terms more precise.

Definition 4.1 (Neighbors)**.** Two sites, s_i and s_j are neighbors if there is an edge $e_{ij} \in \mathcal{E}$. The set of all neighbors of a site, s, is the neighborhood of s and is denoted by $\mathcal{G}_s$.

Definition 4.2 (Neighborhood system)**.** Moreover $\mathcal{G} = \{\mathcal{G}_s | s \in \mathcal{S}\}$ is a neighborhood system for $\mathcal{G}$ if $s \notin \mathcal{G}_s$ and $s \in \mathcal{G}_r \Leftrightarrow r \in \mathcal{G}_s$.

Each site of the graph is associated with a random variable, X_s, that can take any values $x_s \in \Lambda$. A configuration ω describes value assignments at all sites, $\omega = (x_{s_1}, x_{s_2}, ..., x_{s_N})$. The joint probability distribution, $p(X_{s_1} = x_{s_1}, X_{s_2} = x_{s_2}, ..., X_{s_N} = x_{s_N})$, is defined on the set of all possible configurations, $\Omega = \{\omega = (x_{s_1}, x_{s_2}, ..., x_{s_N}) : x_{s_i} \in \Lambda\}$.

Definition 4.3 (MRF)**.** X is a Markov random field (MRF) with respect to $\mathcal{G}$ if

1. for all $\omega \in \Omega$: $p(X = w) > 0$, and
2. for every $s \in \mathcal{S}$ and $\omega \in \Omega$:

   $p(X_s = x_s | X_r = x_r, r \neq s) = p(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s)$.

The second point in Definition 4.3 describes Markov property. A remarkable theorem by Clifford and Hammersley (1971) showed that one can factorize the joint probability density on a Markov random field into a product factors, each defined over a small number of variables. To describe the theorem more precisely, we need to define clique and Gibbs distribution first.

Definition 4.4 (Clique)**.** A subset of sites is a clique, $C \subset \mathcal{S}$, if every pair of sites in C are neighbors. A maximal clique is a clique that cannot be extended by including one more adjacent sites.

Definition 4.5 (Gibbs distribution)**.** A Gibbs distribution is a probability measure on Ω with following form

$$p(\omega) = \frac{1}{Z} \exp -U(\omega)$$

where Z is the normalizing constant (partition function):

$$Z = \sum_{\omega} \exp -U(\omega)$$

and $U(\omega)$ is energy and defined as

$$U(\omega) = -\sum_{\mathcal{T}} V_{\mathcal{T}}(\omega)$$

Each $V_{\mathcal{T}}$ is called an interaction potential and assigns a number to each sub-configuration $\omega_{\mathcal{T}}$.

Now we can describe Hammersely-Clifford theorem (Clifford and Hammersley, 1971).

Theorem 4.6 (Hammersely-Clifford)**.** *X is an MRF with respect to the neighborhood system $\mathcal{G}$ if and only if $p(X = \omega)$ is a Gibbs distribution with potentials over the set of cliques,*

$$p(\omega) = \frac{1}{Z} \exp(-\sum_{C \in C} V_C(\omega))$$

This equivalence is beneficial because it is usually very difficult to specify local characteristics in conditioning neighborhoods. Instead, one can specify potentials over cliques to describe the joint distribution. Interestingly, the Gibbs distribution can also be derived by maximizing entropy. That is, Gibbs distribution has maximal entropy among all probabilty measures on Ω with the same average energy.

When describing images, it is natural to define MRF on a regular lattice, $S = \{s = (i, j) : 0 \leq i, j < M\}$. In this case, we can define n^{th} order neighborhood systems as

$$\mathcal{G}^n = \{\mathcal{G}^n_{(i,j)} : (i, j) \in \mathcal{S}\}$$

$$\mathcal{G}^n_{(i,j)} = \{(k, l) \in \mathcal{S} : (k - i)^2 + (l - j)^2 \leq n\}$$

Figure 4.1 shows first and second order neighborhood systems and their corresponding cliques. In modeling images, choosing the size of the neighborhood is an empirical question. Obviously, reducing the size of the neighborhood decreases the computational cost and alleviates the curse of dimensionality more. However, with too small neighborhoods, it would be impossible to define clique potentials that are complex enough to capture structures of images. Increasing the size of the neighborhood results in more complex maximal cliques, hence potentially more sophisticated clique potentials and eventually better global models. But this comes with a computational cost and gives rise to the same problem MRFs are set to resolve; the curse of dimensionality. Traditionally, more than two order systems were rarely used since the computation required for the complicated energy function would be too expensive (Kato et al., 2012). Ideally, we would like to find the smallest size neighborhoods that can still capture image statistics, along with the most complex interaction potentials. As we will see in this chapter, deep neural network offers a way to achieve this.

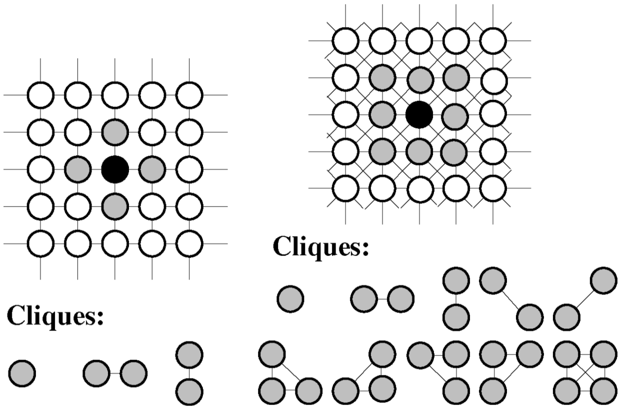


Figure 4.1: Markov neighborhoods on a regular lattice. **Left**. 1st order neighborhood and its cliques. **Right**. 2nd order neighborhood and its cliques (Kato et al., 2012)

4.2 MRF image models for solving inverse problems

A simple MRF is the 2-D Ising model (Kindermann and Snell, 1980) with a 1st order neighborhood system, in which each pixel has four neighbors. Even in such a simple example, it is still difficult to define conditional probabilities given the neighbors. Instead one can easily define potentials over cliques, which consists of two vertical or horizontal adjacent pixels. The energy function is the sum of these potentials defined over pairs of pixels (Kindermann and Snell, 1980). Clearly, the 2-D Ising model is too simplistic for modeling images. Over the years, a variety of MRF models with different neighborhood systems and clique potentials have been proposed. Generally, to test the quality of the models, they are used for solving inverse problems, such as restoration or segmentation, or simply synthesis. The assumption is that a good enough model performs well in these tasks. This section describes example models suggested and tested for these tasks. The general frameworks is as follows: First a model with a particular neighborhood system and local dependencies is suggested. Local dependencies are usually enforced with potentials that encourage some sort of smoothness between neighbors. The parameters of the model are estimated using maximum likelihood estimation (MLE) or expectation-maximization (EM) methods, depending on the problem. Then a sampling method, such as Gibbs sampler, Metropolis Hasting, or loopy belief propagation is used to obtain samples from the model.

An important feature shared across all of these models is their hierarchical nature. Geman and Geman (1984) suggested a two-layer model consists of an intensity process and a line process, where the latter dictates the parameters of the former. In MRF models designed for segmentation, it is common to assume an MRF for hidden variables of regions and an intensity process for pixel values (Bouman and Shapiro, 1994). The role of these types of hierarchies is to give the model the chance to break the smoothness prior enforced by simple local potentials whenever needed, for example around edges. Similarly, in single image super resolution applications, multi-level MRF for gradually coarse grained image intensities has been suggested (Freeman and Liu, 2011). Of particular importance are multi-scale MRF models for restoration and texture synthesis. Multi-scale image decompositions, e.g. wavelet decomposition, offered a mathematical and algorithmic framework better suited for the structural properties of images (Burt and Adelson, 1983; Mallat, 1999). Regardless of the specifics of the hierarchy, a multi-scale representation is necessary for handling of larger structures, and local (Markov) models have captured these probabilistically (e.g., (Buccigrossi and Simoncelli, 1999b; Chambolle et al., 1998; Crouse et al., 1998; Şendur and Selesnick, 2002; Mihçak et al., 1999; Wainwright et al., 2001)).

As we will see below, almost three decades of research on MRFs led to a steady and slow progress in probabilistic image modeling. However, none of these have been able to demonstrate performance comparable to recent deep networks while retaining a local dependency structure (Guth et al., 2022a; Ho et al., 2020). Although we observe impressive performance in some image processing tasks, it is limited to applications that do not rely on long range correlations. For example, Markov random fields have been used to model and synthesize stationary texture images, with conditional dependencies within small spatial regions of the pixel image grid. Beyond stationary textures, however, the chaining of short-range dependencies has been insufficient to capture the complexity of long-range geometrical structures. A question that remains to be answered is whether such failure is inevitable. Or could it be due to certain MRF modeling choices that have been made so far? Often conditioning neighborhoods have been chosen to be too small and clique potentials have been set to simple parametric models mandated by computational limitations. Could less conservative choices result in models that can synthesize non-stationary image structures? We will return to this question later and explore the possibilities provided by the rich expressivity and computational power of DNNs.

4.2.1 Image restoration

Image restoration refers to recovering a clean image from blurry and noisy signal. This is one of the most basic problems in image processing since all optical sensing devices return noisy and blurry images. To remove moderate levels of noise and blur from images, local information is sufficient. This makes restoration a good test-bed for MRF models. Geman and Geman (1984) introduced the first popular MRF for images and specifically designed it for the purpose of restoration. They defined the image as a two layer random process, $X = (F, L)$, where F denotes a Markov random field of observed grayscale discrete intensities, $f \in \Lambda$, and L denotes a Markov random field of unobservable binary line process. X is the union of these two graphs, such that each pixel has four line site neighbors (see Figure 4.2). The goal is to sample from the posterior distribution of X, given the degraded image G. It is worth noting that the assumption is that only intensities are transformed by the degradation and lines are intact and a realization is known a priori. The key in this work, is that the Markov property is not only assumed on the prior, $P(X)$, but also on the posterior, $P(X|G)$. Since the noise is Gaussian, the difference between the energy function of the posterior and prior is a quadratic term from the likelihood. The potentials, V_C, are chosen ad hoc (instead of MLE) such that they penalize intensity gap between neighbors in cliques. Moreover, the line process breaks the smoothness prior of the intensity process, such that if $l \neq 0$ then intensity gap between neighbors is not penalized. The neighborhood system is of 2^{nd} order, i.e. 3×3 neighborhoods. In order to make the model work reasonably with these simple assumptions, the test images are very simple and consist of only 5 levels of intensities. Finally to

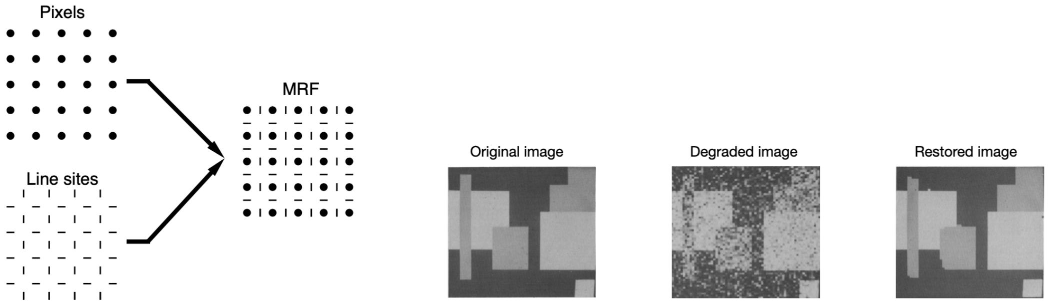


Figure 4.2: Markov Random Field for restoration **Left.** Intensity and line process and their union. **Right.** Example degraded image along with the performance of the model. (Geman and Geman, 1984)

sample from the posterior, they device a Gibbs sampler with annealing schedule which updates one coordinate at a time. The goal of annealing is to gradually decreasing the temperature to avoid local maxima. In the initial iterations, the high temperature smooths out the density and makes it easy to escape local maxima. As temperature decreases, the density becomes more peaked and the solution gets pinned down with reduced chaos. Figure 4.2 shows as example clean image along with degraded and restored versions. Geman and Geman laid out a thorough framework for MRFs in image modeling. The main shortcomings were the oversimplification of clique potentials as well as assumed realization of the line process.

As mentioned before, various MRF models were proposed for restoration. Portilla et al. (2003b) proposed Scale Mixtures of Gaussians(GSM), an MRF model with impressive and state-of-the-art performance in denoising. First, the image was transformed to a multi-scale over-complete wavelet representation. The local neighborhoods were defined across space and scale as shown in Figure 4.3. This choice was based on the empirical analysis suggesting a strong dependency between child and parent coefficients, Figure 4.3. Then the wavelet coefficients were decomposed into the product of two random variables: a Gaussian vector and a hidden positive scalar multiplier, $x = \sqrt{z}u$. Hidden variable z modulates the local variance of the coefficients in the neighborhood, and is thus able to account for the empirically observed correlation between the coefficient amplitudes. The introduction of z is one of the key reasons in the success of the model, because it makes the interaction potential adaptive to each neighborhood. In this work, similarly to (Geman and

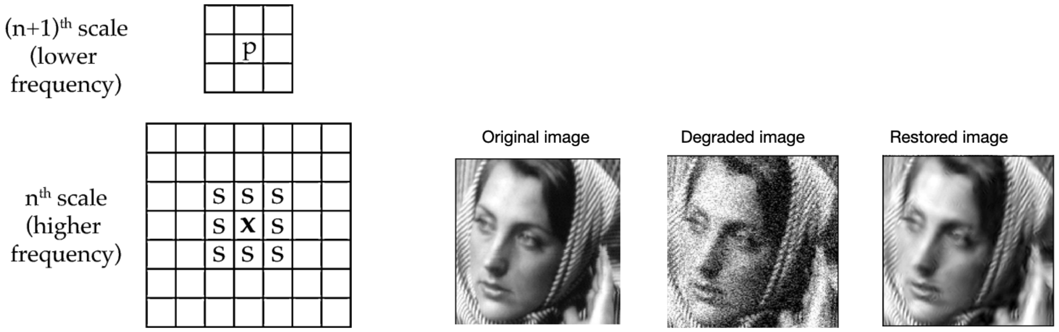


Figure 4.3: Markov neighborhoods in GSM model. **Left.** Local neighborhoods across scale and space in GSM. **Right.**Example degraded image along with the performance of the model. (Portilla et al., 2003b)

Geman, 1984), the goal is to estimate and sample from the posterior distribution given the noisy coefficients, $p(x_c|y)$, where x_c is the center true coefficient in a 3×3 neighborhood, and y is the vector of noisy coefficients in the neighborhood. A key difference, however, is that rather than sampling from the posterior, the algorithm returns the mean of the posterior (i.e. the minimizer of mean squared error) as the optimal denoised coefficient. This offers a computational advantage by removing the need for explicitly expressing the posterior. Instead, they only need to estimate $p(z|y)$ from data which is much simpler. This method is truly Bayesian, because instead of a MAP solution for the posterior, it marginalizes over all values of z,

$$\mathbb{E}\{x_c|y\} = \int_0^\infty p(z|y)\mathbb{E}\{x_c|y, z\}dz$$

Although this approach has a computational advantage, which is likely responsible for its empirical success, the absence of explicit clique potentials prevent us from obtaining a global Gibbs distribution. Nevertheless, this model relies on the Markov property that limits the estimation of a coefficient to conditioning on a small neighborhood, and offers a hidden MRF. This compromise can be viewed as a trade-off between the explicit parameterization of the clique potentials and the empirical performance. We will see more examples of this trade-off in the next sections. Figure 4.3 shows an example denoising result from this algorithm.

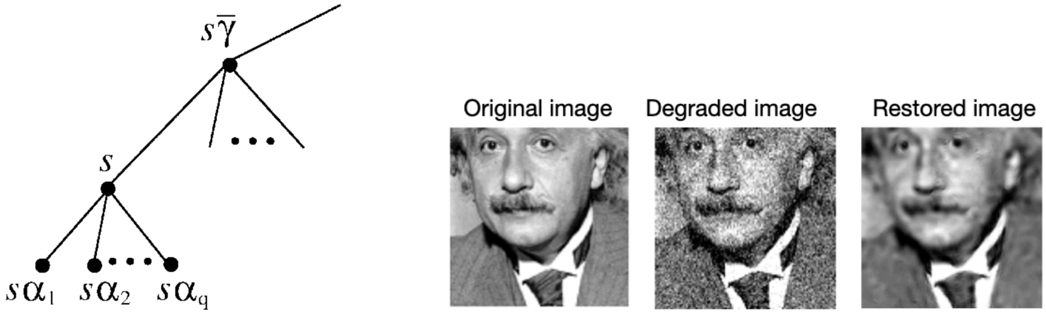


Figure 4.4: Hierarchical GSM. **Left.** A segment of a q-adic tree, with the unique parent $s\bar{y}$ and children $s\alpha_1, s\alpha_2, ...$ corresponding to node s. **Right.** Example degraded image along with the performance of the model. (Wainwright et al., 2000)

Wainwright et al. (2000) proposed a variant of GSM model designed to express a global Gibbs distribution. In order to do that, they reduced the neighborhoods to only two coefficients, parent and child, eliminating all the spacial coefficients. This choice of the neighborhood results in a quad-tree, rising from multi-scale wavelet representation, and allows an explicit definition of GSM clique potentials. The non-loopy tree structure allows for an exact estimation of the posterior with belief propagation algorithm. Figure 4.4 shows an example denoising performance based on this model. As clear by the example, the parametric expression of the global model comes at the cost of reduced performance.

4.2.2 Image segmentation

Image segmentation is a difficult and mostly ill-defined problems in computer vision. Given an image,Y, the goal is to assign labels, X, to pixels, where values of labels, $x \in L$ consist a discrete set, and each value represents a region or segment in the image. The goal is to find optimal labels given the image, that is, to find the maximum of the posterior, $p(X|Y)$. According to the Bayes rule, the MAP estimate is equivalent to the maximiser of likelihood times prior, $p(Y|X)p(X)$. Markov random fields are suitable for modeling the prior of the hidden labels, because Markov property is a reasonable assumption; the probability of a given label is independent from faraway labels, conditioned on the surrounding labels. Thus, the posterior can be defined with a layered or hierarchical MRF (Figure 4.5). Generally the framework of the problem is as follows. Take a simple first order neighborhood system on the X graph. Such local interactions produce doubleton cliques that consists of only two sites. A simple potential on such cliques comes from a smoothness prior similarly to Ising model or the restoration problem. The potential can be a positive fixed value for different labels in the doubleton, resulting in higher energy and lower probability. Conversely, the clique potential for similar values in the doubleton can be set to a fixed negative value to maximize probability.

$$V_C = \beta.\delta(x_{s_1}, x_{s_2}) = \begin{cases} +\beta, & \text{if } x_{s_1} \neq x_{s_2} \\ -\beta, & \text{otherwise} \end{cases}$$

Additionally, the likelihood model, $p(Y|X)$, can be represented by a mixture of Gaussians. The likelihood tells us the probability of a pixel belonging to a region with a particular mean and variance. Generally, likelihood model determines the behavior of the image (e.g., texture, gray scale, color, or multispectral values). If the parameters of the likelihood model are known apriori, then the solution comes down to finding the maximum of the posterior. It is common to relax the MAP problem with sampling due to non-convex nature of the optimization. To sample from larger maxima, usually an annealed MCMC method is used. This method can be implemented only if the likelihood model parameters are known, for example if the mean and variances of the Gaussians can be estimated from a set of training data. When such information is not available, the MAP solution is replaced with an EM solution, where the parameters of the model and the minimum cost label assignments are estimated from data iteratively in alternating steps.

While the above-mentioned framework describes the backbone of most MRF segmentation methods, many variations have been proposed to improve the results. For example, Kato (2008) suggests using a reversible jump Markov chain Monte Carlo (RJMCMC) method that makes it possible for samplers to jump between parameter subspaces of different dimensionality. Almost

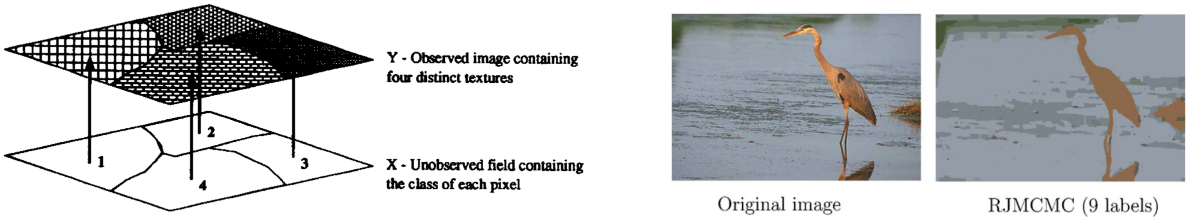

Y - Observed image containing
four distinct textures
X - Unobserved field containing
the class of each pixel
1
2
3
4
Original image
RJMCMC (9 labels)

Figure 4.5: Markov random field for segmentation. **Left**. Structure of the doubly stochastic random field used in segmentation. The behavior of the image (e.g., texture, gray scale, color, or multispectral values) given the class labels is defined by the likelihoof, $p(Y|X)$. Prior information is contained in the distribution of class labels $p(X)$.(Bouman and Shapiro, 1994) . **Right.** Example segmentation of image (Kato et al., 2012)

all other parts of the algorithm are the same as what was described above. Figure 4.5 shows an example image segmented by this method in a fully unsupervised way (EM). Other works by Bouman and Liu (1988) assumed a mixture of multi-dimensional Gaussians as their texture model which was used as the likelihood. More importantly, they implemented a multi-resolution segmentation algorithm by first segmenting the image at coarse resolution and then progressing to finer resolutions until each individual pixel is labeled. Segmentation at each resolution is performed by maximizing the posterior subject to the label constrains. The cascade of these iterative MAP estimations turns out to be computationally efficient. Also, choosing more expressive probabilities families for $p(x)$ is another way to improve the segmentation model. For example, (Bouman and Sauer, 1994) replaced the discrete Ising model smoothness prior with a continuous generalized Gaussian. Gaussian potentials do not model edges well. A number of None-Gaussian potentials for pair-wise cliques have been proposed. Examples are shown in Figure 4.6. Although these potentials have the advantage of creating sharp edges, it is very difficult to compute MAP estimates when using them.

4.2.3 Texture synthesis

Markov random fields have been used to model stationary texture images, with conditional dependencies within small spatial regions of the image grid. MRF modeling has proved to be

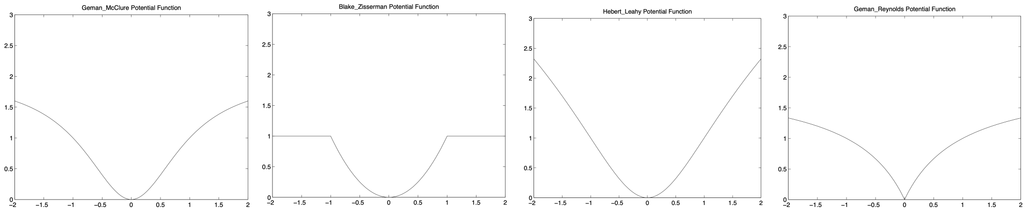


Figure 4.6: Non-Gaussian pair-wise clique potentials, $V_c(\Delta)$, where $\Delta = (x_{s_1} - x_{s_2})/\sigma^2$. From left to right: $\frac{\Delta^2}{1+\Delta^2}$ (Geman et al., 1992), $\min\{\delta^2, 1\}$ (Blake, 1989), $\log(1+\Delta^2)$ (Hebert and Leahy, 1989), $\frac{|\Delta|}{1+|\Delta|}$ (Geman and Reynolds, 1992). All of these penalize difference between label values up to a threshold, beyond which the potential either does not increase or increase with a smaller rate. This construction model edges better while still promotes smoothness

insufficient for non-stationary images with global structure. But textures consists of local and repetitive patterns which makes them a natural candidate for MRF modeling and inference. In this section I describe an MRF texture model (Paget and Longstaff, 1998) that takes a step in the direction of capturing more complex local interactions than the conventional parametric MRF models. It has been long understood that higher order models are required for texture synthesis. While MRFs theoretically have the required statistical order, the computational limitations usually comes in the way of choosing larger neighborhoods and more complex potentials. (Paget and Longstaff, 1998) proposed a non-parametric texture MRF model to allow for a more complex local conditional probability density function (LCPDF) over neighborhoods as larger as 9 × 9. The LCPDF is separately learned for each new class of texture by a histogram estimation. Assuming spacial stationary,

$$P(x_s | x_r, r \in \mathcal{N}_s) = \frac{F(L_0 = x_s, L_{n_r} = x_r, r \in \mathcal{N}_s)}{\sum_{L_0 \in \Lambda} F(L_0 = x_s, L_{n_r} = x_r, r \in \mathcal{N}_s)}$$

Where the index n_r denotes the relative position of site r to p. After building the multi-dimensional histogram, a multi-dimensional Gaussian Parzen window is used to smooth it out. The size of the window parameter is set to an optimal value for LCPDE. In order to alleviate the curse of dimensionality in estimating LCPDE for larger neighborhoods, a multi-scale framework is introduced. The multi-scale representation comes from decimating the original image, Fig 4.7.

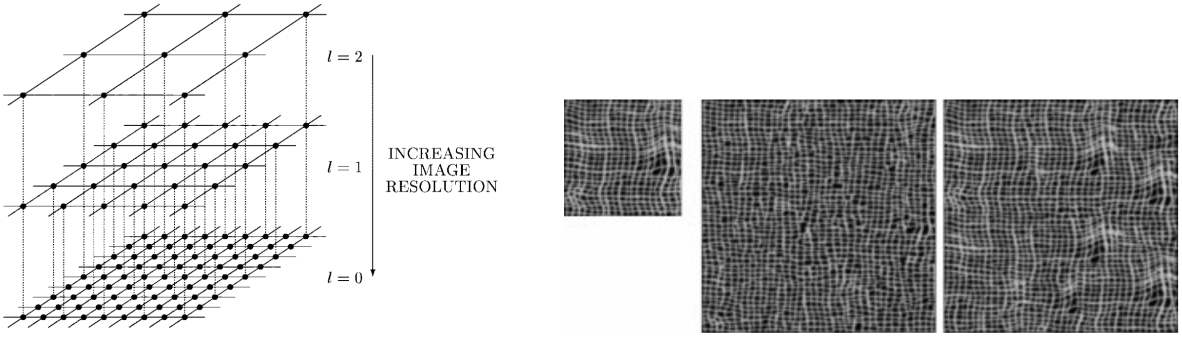


Figure 4.7: Markov random field for texture synthesis. **Left.** Grid organization for MR via decimation. **Right.** Original image. Synthesized textures—using neighborhood order of 8 (i.e 5×5). Synthesized textures—using neighborhood order of 18 (i.e 9×9)

Each scale is a new MRF with a lower dimensional lattice. The neighborhoods are redefined to adjust for the smaller lattice. Starting from the top of the pyramid, a set of i.i.d sites are chosen. Then LCPDF for x_s is estimated in parallel using the above question and Parzen smoothing. Then a new set of sites are chosen and their value is estimated by their LCPDF, as in Gibbs sampler. In order to further reduce curse of dimensionality, the authors introduced "pixel temperature", a local annealing procedure that assigns a temperature or confidence score, t, to each x_s. When $t = 0$ there is no uncertainty about the value of the pixel and the update can stop. All pixel temperatures start at $t = 1$ and are updated in each iteration according to an update rule. t reaches zero only when all the neighbors have achieved zero. When all temperatures in the grid are zero, the sampler stops and the coarse image is passed to the next step. The critical point is that the values and pixel temperatures associated with pixel are fixed and don't get updated even after being passed down to the new finer level. In addition to lifting the curse of dimensionality, pixel temperature prevents formation of artifacts. Figure 4.7 shows example textures synthesized using this method. The impressive performance of this model indicates that larger neighborhoods and more flexible and complex local interaction laws are necessary for a good MRF model. The downside of this model however, is that it is not clear how one can define a valid joint distribution using the LCPDF.

4.3 Markov Wavelet Conditional Models

This section introduces a relatively low-dimensional model class as a product of Markov conditional probabilities over multi-scale wavelet coefficients. In the next section, we show that this model can be successfully implemented using convolutional neural networks.

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, 1999), to iteratively compute wavelet coefficients (see Figure 4.8). Let $\boldsymbol{x}_0$ be an image of $N \times N$ pixels. For each scale $j > 1$, the operator W decomposes $\boldsymbol{x}_{j-1}$ into:

$$W\boldsymbol{x}_{j-1} = (\bar{\boldsymbol{x}}_j, \boldsymbol{x}_j),$$

where $\boldsymbol{x}_j$ is a lower-resolution image and $\bar{\boldsymbol{x}}_j$ is an array of three wavelet coefficient images, each with dimensions $N/2^j \times N/2^j$, as illustrated in Figure 4.8. The inverse wavelet transform iteratively computes $\boldsymbol{x}_{j-1} = W^T(\bar{\boldsymbol{x}}_j, \boldsymbol{x}_j)$.

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

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

This is performed J times, so that the lowest resolution image $\boldsymbol{x}_J$ is small enough, which yields:

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

The conditional distributions $p(\bar{\boldsymbol{x}}_j|\boldsymbol{x}_j)$ 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:

$$p(\bar{\boldsymbol{x}}_j|\boldsymbol{x}_j) = \mathcal{Z}_j(\boldsymbol{x}_j)^{-1}\, e^{-E_j(\bar{x}_j|\boldsymbol{x}_j)}, \tag{4.2}$$

where $\mathcal{Z}(\boldsymbol{x}_j)$ is the normalization constant for each $\boldsymbol{x}_j$. 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 $p(\bar{\boldsymbol{x}}_j|\boldsymbol{x}_j)$ is a conditional Markov random field. That is, the probability distribution of a wavelet coefficient of $\bar{\boldsymbol{x}}_j$ conditioned on values of $\boldsymbol{x}_j$ and $\bar{\boldsymbol{x}}_j$ in a restricted spatial neighborhood is independent of all coefficients of $\bar{\boldsymbol{x}}_j$ and $\bar{\boldsymbol{x}}$ outside this neighborhood (see Figure 4.8). 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

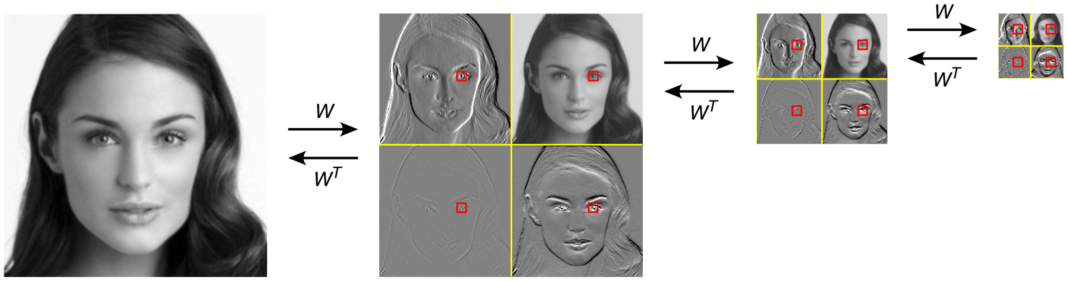


Figure 4.8: Markov wavelet conditional model structure. At each scale j, an orthogonal wavelet transform W decomposes an image $\boldsymbol{x}_{j-1}$ into three wavelet channels, $\bar{\boldsymbol{x}}_j$, containing vertical, horizontal, and diagonal details, and a low-pass channel $\boldsymbol{x}_j$ 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 $\bar{\boldsymbol{x}}_j$ (here, centered on the left eye), conditioned on values of $\boldsymbol{x}_j$ and $\bar{\boldsymbol{x}}_j$ in a local spatial neighborhood (red squares), is independent of all coefficients of $\bar{\boldsymbol{x}}_j$ outside this neighborhood.

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):

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

Computing the score (4.3) is equivalent to specifying the Gibbs model (4.2) without calculating the normalization constants $\mathcal{Z}(\boldsymbol{x}_j)$, since these are not needed for noise removal, super-resolution or image synthesis applications.

4.4 Score-based Markov Wavelet Conditional Models

Score-based diffusion models have produced impressive image generation results (e.g., (Guth et al., 2022a; Ho et al., 2020; 2022; Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Song et al., 2021b)). 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.

To model the conditional wavelet distribution $p(\bar{x}_j|x_j)$, Guth et al. (2022b) 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)). 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. (2.3), 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 4.7. *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 $1 \leq j \leq J$, 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 4.7, this defines the Markov conditioning neighborhoods of the learned model. The effect of the RF size is examined in the numerical experiments of Section 4.5.

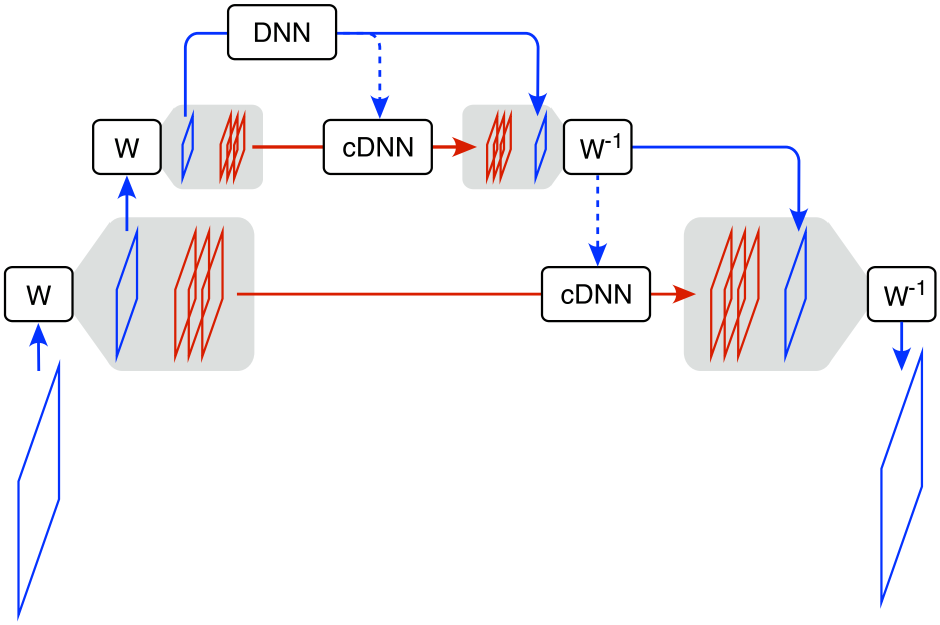


Figure 4.9: Wavelet conditional denoiser architecture used to estimate the score (illustrated for a two-scale decomposition). The input noisy image $\boldsymbol{y}$ (lower left) is decomposed by recursive application of a fast orthogonal wavelet transform W into successive low-pass images $\boldsymbol{y}_j$ (blue) and three wavelet detail images $\bar{\boldsymbol{y}}_j$ (red). The coarsest low-pass image $\boldsymbol{y}_J$ is denoised using a CNN with a global receptive field to estimate $\hat{\boldsymbol{x}}_J$. At all other scales, a local conditional CNN (cCNN) estimates $\hat{\bar{\boldsymbol{x}}}_j$ from $\bar{\boldsymbol{y}}_j$ conditioned on $\hat{\boldsymbol{x}}_j$, from which W^T recovers $\hat{\boldsymbol{x}}_{j-1}$.

Parameterizing the score with a convolutional network further implies that the conditional probability $p(\bar{\boldsymbol{x}}_j|\boldsymbol{x}_j)$ is stationary on the wavelet sampling lattice at scale j. Despite these strong simplifications, we shall see that these 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 $\boldsymbol{x}_J$, and thus does not enforce local Markov conditioning nor stationarity. The product density of eq. (4.1) is therefore not stationary.

4.5 Markov Wavelet Conditional Denoising

We now evaluate our Markov wavelet conditional model on a denoising task. We use the trained CNNs to define a multi-scale denoising architecture, illustrated in Figure 4.9. The wavelet transform of the input noisy image $\boldsymbol{y}$ is computed up to a coarse-scale J. The coarsest scale image is denoised by applying the denoising CNN learned previously: $\hat{\boldsymbol{x}}_J = f_J(\boldsymbol{y}_J)$. Then for $J \geq j \geq 1$, we compute the denoised wavelet coefficients conditioned on the previously estimated coarse image: $\hat{\bar{\boldsymbol{x}}}_j = f(\bar{\boldsymbol{y}}_j, \hat{\boldsymbol{x}}_j)$. We then recover a denoised image at the next finer scale by applying an inverse wavelet transform: $\hat{\boldsymbol{x}}_{j-1} = W^T(\hat{\bar{\boldsymbol{x}}}_j, \hat{\boldsymbol{x}}_j)$. At the finest scale we obtain the denoised image $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_0$.

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: $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \sum_{j=1}^{J-1} \|\bar{\boldsymbol{x}}_j - \hat{\bar{\boldsymbol{x}}}_j\|^2 + \|\boldsymbol{x}_J - \hat{\boldsymbol{x}}_J\|^2$. 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.2 provides architecture and training details.

Figure 4.10 shows that the multi-scale 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 4.10, 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, multi-scale 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 4.10, 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 multi-scale 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 multi-scale 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.

To further illustrate this point, consider the denoising examples shown in Figure 4.11. Since all denoisers are bias-free, they are piecewise linear (as opposed to piecewise affine), providing some interpretability, as discussed in Chapter 2. 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 multi-scale 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 multi-scale 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

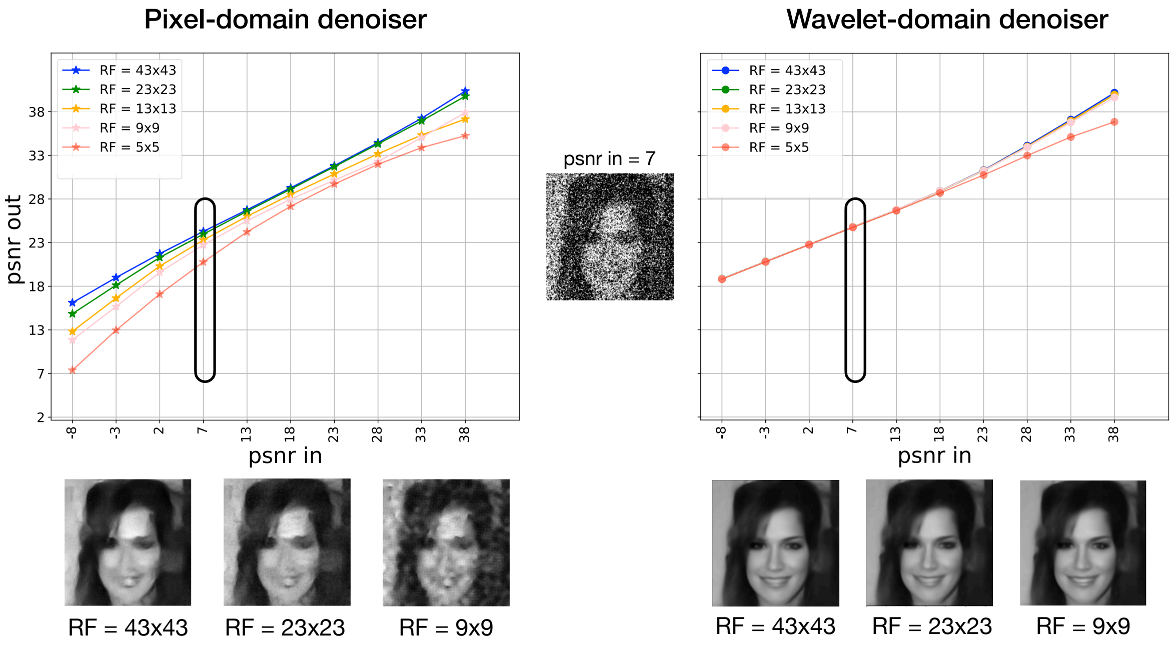


Figure 4.10: Comparison of denoiser performance on pixel vs wavelet domain. Test images are of resolution of 160×160 from the CelebA dataset. Each panel shows the error of the denoised image as a function of the noise level, both expressed as the peak signal-to-noise ratio (PSNR). **Left:** Conventional CNN denoisers with different RF sizes. The blue curve shows performance of BF-CNN. The rest are BF-CNN variants with smaller RF obtained from setting some intermediate filter sizes to 1×1. Also shown are the outputs of three denoisers with varying RF on a test image. **Right:** Multi-scale denoisers, as depicted in Figure 4.9, with different cCNN RF sizes. Performance us almost the same on the test image regardless of denoiser's RF. Note that the low-pass denoiser RF is 40×40 in all cases, and thus covers the entire low-pass band.

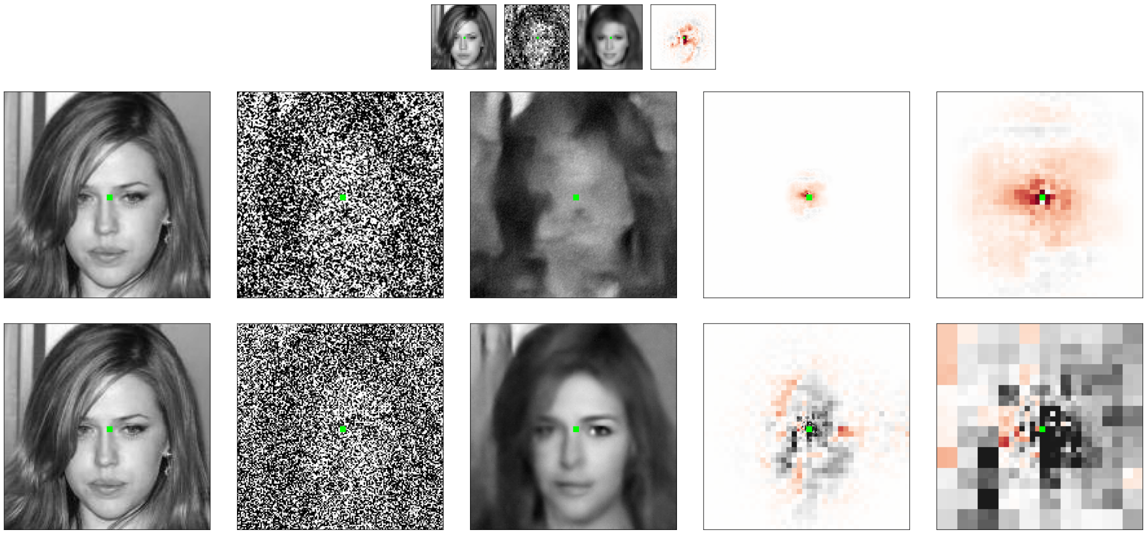


Figure 4.11: Denoising examples. Each row shows clean image, noisy image, denoised image, and the adaptive filter (one row of the Jacobian of the end-to-end denoising transformation) used by the denoiser to estimate a specific pixel, indicated in green. The heat-map ranges from red for most negative 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 multi-scale denoiser of Figure 4.9. The denoiser uses a 40×40 RF for the coarsest scale, and 13×13 RFs for conditional denoising of subsequent finer scales.

resolution, over ever smaller RFs relative to the coefficient 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.

4.6 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 $\boldsymbol{x}_J$ is used to conditionally generate wavelet coefficients $\bar{\boldsymbol{x}}_J$ from the conditional distribution $p(\bar{\boldsymbol{x}}_J|\boldsymbol{x}_J)$. An inverse wavelet transform next recovers a higher-resolution image $\boldsymbol{x}_{J-1}$ from both $\boldsymbol{x}_J$ and $\bar{\boldsymbol{x}}_J$. 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 $\boldsymbol{x}$ from the full-resolution image distribution conditioned on the starting low-resolution image $p(\boldsymbol{x}|\boldsymbol{x}_J)$. $\boldsymbol{x}$ is thus a stochastic super-resolution estimate of $\boldsymbol{x}_J$.

To draw samples from the distributions $p(\bar{\boldsymbol{x}}_j|\boldsymbol{x}_j)$ implicitly embedded in the wavelet conditional denoisers, we use the anti-diffusion algorithm (SISA) introduced in Chapter 2. Extension to the conditional case is straightforward. The sampling algorithm is detailed in Algorithm 2. All the cCCN 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. For the examples in Figures 4.12, C.1 and 4.13, we chose $h = 0.01$, $\sigma_0 = 1$, $\beta = 0.1$ and $\sigma_\infty = 0.01$. 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 $\nabla \log p(\boldsymbol{x})$. Given the same low-pass image $\boldsymbol{x}_J$, we can generate samples from $p(\boldsymbol{x}|\boldsymbol{x}_J)$ by viewing this as sampling from the image distribution of $\boldsymbol{x}$ constrained by a linear measurements $\boldsymbol{x}_J$. This method is described in more detail in Chapter 5.

Algorithm 2: Wavelet Conditional Synthesis

```
parameters: number of scales $J$, low-pass image $\boldsymbol{x}_J$, conditional denoisers $(f_j)_{1 \leq j \leq J}$, step size $h$,
  initial noise level $\sigma_0$, final noise level $\sigma_L$
for $j \in \{J, \ldots, 1\}$ do
    Wavelet conditional sampling:
    $\bar{\boldsymbol{x}}_j \leftarrow \text{DrawSample}(f_j(\cdot, \boldsymbol{x}_j), h, \sigma_0, \sigma_L)$
    Wavelet reconstruction:
    $\boldsymbol{x}_{j-1} \leftarrow W^T(\bar{\boldsymbol{x}}_j, \boldsymbol{x}_j)$
end
return $\boldsymbol{x}_0$
```

Figure 4.12 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, $\nabla_{\boldsymbol{x}_J} \log p(\boldsymbol{x}_J)$, 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

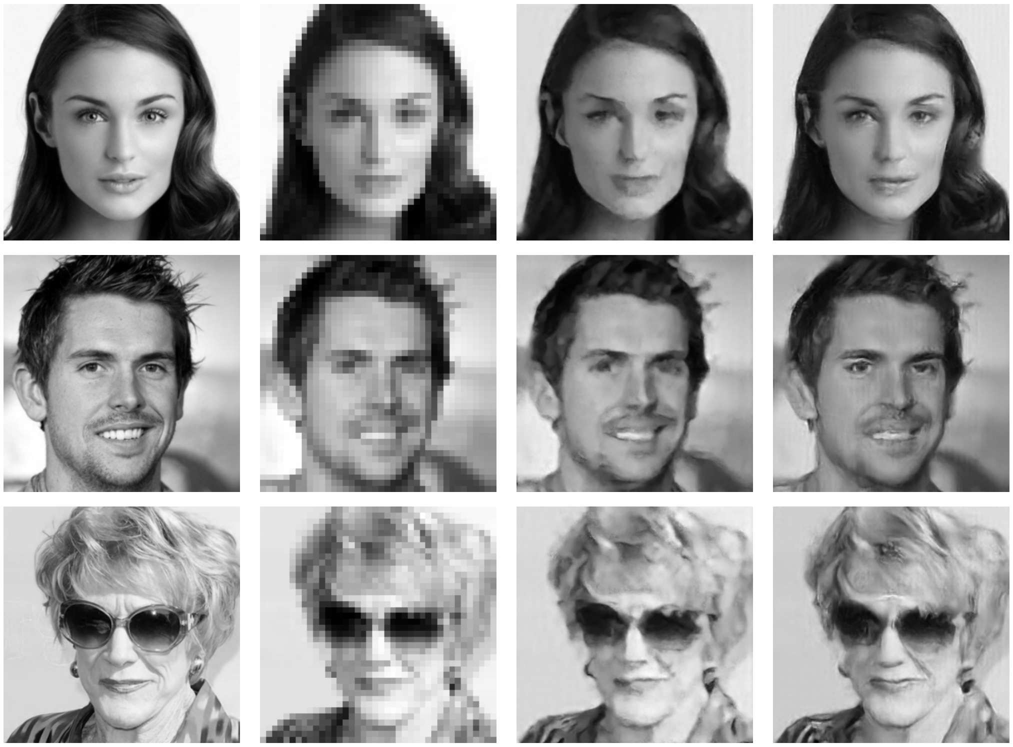


Figure 4.12: Super-resolution examples. **Column 1**: original images (320 × 320). **Column 2**: Low-pass image of a 3-stage wavelet decomposition (downsampled to 40 × 40) expanded to full size for viewing. **Column 3**: Conditional generation of full-resolution images using CNN denoiser with RF of size 43 × 43. **Column 4**: Coarse-to-fine conditional generation using the multi-scale cCNN denoisers, each with RFs of size 13 × 13. Additional examples are shown in Appendix C.3.

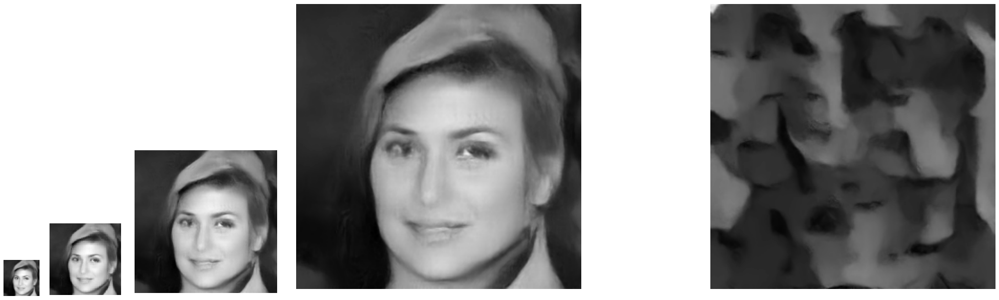


Figure 4.13: Image synthesis. **Left four images:** Coarse-to-fine synthesis, achieved by sampling the score learned for each successive conditional distribution. Synthesized images are shown at four resolutions, from coarse-scale only (leftmost, 40×40) to the finest scale (rightmost, 320×320). Conditional 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.

successive scale are generated by drawing a sample using the cCNN conditioned on the previous scale.

The first (smallest) image in Figure 4.13 is generated from the low-pass CNN (see Chapter 2 for SISA 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 multi-scale super-resolution synthesis algorithm described above. The next three images in Figure 4.13 show successively higher resolution images generated in the procedure. For comparison, the last image in Figure 4.13 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.

4.7 Discussion

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 (Guth et al., 2022b; Ho et al., 2020), which also compute conditional scores at each scale.

Our experiments were performed on cropped and centered face images, which present a particular challenge given their obvious non-stationarity. Although the cCNN networks in our model are fully convolutional, the conditional models are only approximately stationary, because of the zero-padded boundary handling. The influence of the boundary handling grows for the coarser scales, giving rise to the non-stationary capabilities of the full model. In particular, the terminal low-pass CNN is fully non-stationary. Furthermore, the overall model has a "foveated" structure. The RF of the finer scales is smaller compared to the size of the entire grid. In the coarser scales, the ratio of the RF size to the grid size grows exceeding 1 for the low-pass grid, resulting in a global RF. The gradual transition from local stationary to global non-stationary models across scale is the key reason for success of the overall model.

Previously, diffusion models (Guth et al., 2022b; Ho et al., 2022) have used coarse-to-fine strategies, generating a sequence of images of increasing resolution, each seeded by its predecessor. None of these models, however, 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, as opposed to ours which uses 600k parameters for the low-pass CNN and 200k for the cCNNs), with global receptive fields. As a result, these models require much larger training sets to enter the generalization regime. The success of the multi-scale conditional model suggests that there is a lot of redundancy in finer components of the image, which can be captured with smaller models. Additionally, our local conditional Markov assumptions provide a step towards the goal of making explicit the probability model and its factorization into low-dimensional components.

Another potential direction to explore is assuming a scale invariance property. 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.

5 Using learned image densities to solve linear inverse problems

In the previous chapter, we showed that the learned prior embedded in the denoiser can be used to study properties of image distribution such as locality and stationarity. In this chapter, we aim to utilize the prior more directly to solve linear inverse problems. Many applications in signal processing can be expressed as linear inverse problems - deblurring, super-resolution, estimating missing pixels (e.g., inpainting), and compressive sensing are all examples. Given a set of linear measurements of an image, $x^c = M^T x$, where M is a low-rank measurement matrix, one attempts to recover the original image. A prior probability model can provide a universal substrate for solving inference problems. This is in contrast to mapping-based approach where the solution is intertwined with the task for which the network is optimized, so it requires training a separate network for each new application.

In Section 2.2, we developed a stochastic iterative score ascent (SISA) algorithm for obtaining a high-probability sample from $p(x)$. Here, we generalize SISA algorithm to solve for a high-probability sample from the conditional density $p(x|M^T x = x^c)$. The resulting procedure generates high-probability samples from the prior conditioned on the measurements, thus pro-

Some of the results in this chapter are published in (Kadkhodaie and Simoncelli, 2020) and (Kadkhodaie and Simoncelli, 2021a). The code to reproduce the results is available at https://github.com/LabForComputationalVision/universal_inverse_problem

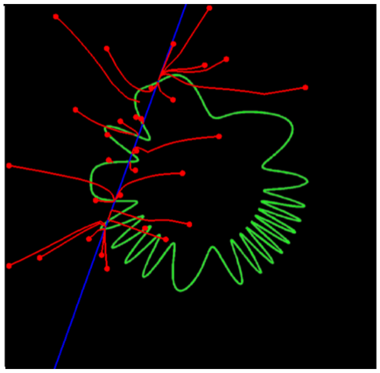


Figure 5.1: Two-dimensional simulation/visualization of constrained sampling. Only points lying at the intersection of manifold (green curve) and constraint hyperplane (blue line, represented by low-rank matrix M) are valid samples.

viding a general stochastic solution for any linear inverse problem. Geometrically, constrained sampling corresponds to drawing points which sit at the intersection of the image manifold and the constrained hyperplane (see Fig. 5.1). Note that for these problems, the injection of noise ($\beta < 1$) is particularly important, because points on the intersection are not necessarily the closest points to the initial image.

We demonstrate that our method produces visually high-quality results in recovering missing pixels, and state-of-the-art levels of unsupervised performance on superresolution, deblurring and compressive sensing.

5.1 Constrained sampling algorithm

Consider the distribution of a noisy image, y, conditioned on the linear measurements, $x^c = M^T x$. Without loss of generality, we assume the columns of the matrix M are orthogonal unit vectors.[1] We project y onto two complementary subspaces spanned by the measurement matrix and its orthogonal complement $\bar{M}$. We write the conditional density of the noisy image conditioned

[1] If not, we can re-parameterize to an equivalent constraint using the SVD. In this case, M is the pseudo-inverse of M^T, and the matrix MM^T projects an image onto the measurement subspace.

on the linear measurement as

$$p(y|x^c) = p(y^c, y^u|x^c) = p(y^u|y^c, x^c)p(y^c|x^c) = p(y^u|x^c)p(y^c|x^c)$$

where $y^c = M^T y$, $y^u = \bar{M}^T y$. The last equality is obtained by considering that y^c is equal to x^c plus independent Gaussian noise. So given x^c, y^c does not provide any additional information about y^u. That is, y^u is independent of y^c when conditioned on x^c. As with SISA algorithm of Section 2.2, we wish to obtain a local maximum of this function using stochastic coarse-to-fine gradient ascent. Applying the operator $\sigma^2 \nabla \log(\cdot)$ yields

$$\sigma^2 \nabla_y \log p(y|x^c) = \sigma^2 \nabla_y \log p(y^u|x^c) + \sigma^2 \nabla_y \log p(y^c|x^c)$$

The second term is the gradient of the log of the observation noise distribution that lies within the measurement subspace (column space of M). For Gaussian noise with variance σ^2, it reduces to $M(y^c - x^c)$. The first term is the gradient of log of the noisy image distribution in the subspace orthogonal to the measurement subspace, conditioned on the measurements. This can be computed by projecting the measurement subspace out of the full gradient given by the denoiser residual. Specifically, we project $f(y)$ onto the orthogonal complement of M using the matrix $I - MM^T$. Combining these gives:

$$\begin{aligned} \sigma^2 \nabla_y \log p(y) &= (I - MM^T)\sigma^2 \nabla_y \log p(y) + M(x^c - y^c) \\ &= (I - MM^T) f(y) + M(x^c - M^T y). \end{aligned} \tag{5.1}$$

Thus, we see that the gradient of the conditional density is partitioned into two orthogonal components, capturing the gradient of the (log) noisy density, and the deviation from the constraints, respectively. To draw a high-probability sample from $p(x|x^c)$, we use the same algorithm described in Section 2.2, substituting Eq. (5.1) for the deterministic update vector, d_t (see Algorithm

3 and Figure 5.2). Note that without any measurements (i.e., $M = 0$) Algorithm 3 reduces to Algorithm 1.

Algorithm 3: Stochastic iterative score ascent (SISA) method for sampling from $p(x|M^T x = x^c)$, based on the residual of a denoiser, $f(y) = \hat{x}(y) - y$. Note: e is an image of ones.

```
parameters: $\sigma_0$, $\sigma_L$, $h_0$, $\beta$, $M$, $x^c$
initialization: t=1; draw $y_0 \sim \mathcal{N}(0.5(I - MM^T)e + Mx^c,\ \sigma_0^2 I)$
while $\sigma_{t-1} \geq \sigma_L$ do
    $h_t = \frac{h_0 t}{1+h_0(t-1)}$;
    $d_t = (I - MM^T)f(y_{t-1}) + M(x^c - M^T y_{t-1})$;
    $\sigma_t^2 = \frac{||d_t||^2}{N}$;
    $\gamma_t^2 = \left((1 - \beta h_t)^2 - (1 - h_t)^2\right) \sigma_t^2$;
    Draw $z_t \sim \mathcal{N}(0, I)$;
    $y_t \leftarrow y_{t-1} + h_t d_t + \gamma_t z_t$;
    $t \leftarrow t + 1$
end
```

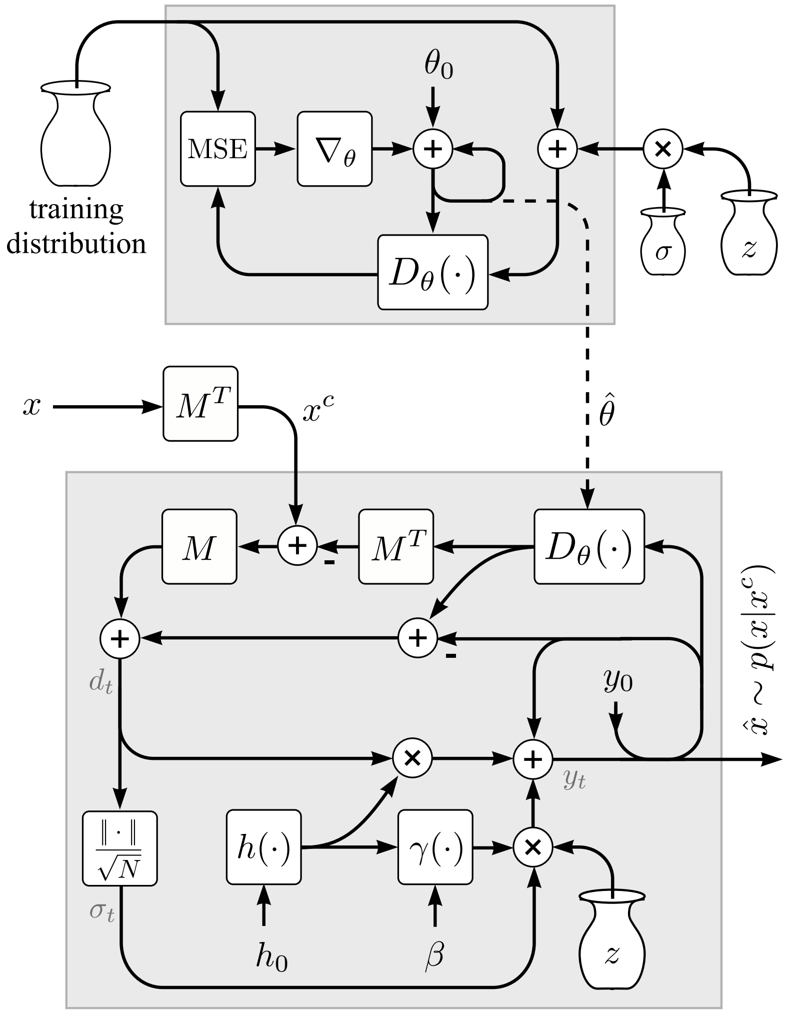


Figure 5.2: Block diagrams for denoiser training, and Universal Inverse Sampler. **Top:** A parametric blind denoiser, $D_\theta(\cdot)$, is trained to approximate $\hat{x}(y)$ by minimizing mean squared error when removing additive Gaussian white noise (z) of varying amplitude (σ) from images drawn from a training distribution. The trained denoiser parameters, $\hat{\theta}$, constitute an implicit model of this distribution. **Bottom:** The trained denoiser is embedded within an iterative computation to draw samples from this distribution, starting from initial image y_0, and conditioned on a low-dimensional linear measurement of a test image: $\hat{x} \sim p(x|x^c)$, where $x^c = M^T x$. If measurement matrix M is empty, the algorithm draws a sample from the unconstrained distribution. Parameter $h_0 \in [0, 1]$ controls the step size, and $\beta \in [0, 1]$ controls the stochasticity (or lack thereof) of the process.

5.2 Linear inverse examples

We evaluate our method on five linear inverse problems. The same algorithm and parameters are used on all problems - only the measurement matrix M and measured values $M^T x$ are altered. In particular, as in section 2.2.2, we used BF-CNN, and chose parameters $\sigma_0 = 1, \sigma_L = 0.01, h_0 = 0.01, \beta = 0.01$. For each example, we show original images (x), the direct least-squares reconstruction ($MM^T x$), and restored images. Subjective assessment of perceptual quality is particularly important in cases where the measurement matrix is of very low rank, and the distribution of solutions is diverse (the synthesis examples of the previous section correspond to the limiting case, with measurements of rank zero). We also provide numerical comparisons with other unsupervised methods, in terms of both PSNR and SSIM (an approximate measure of perceptual quality). Since our method is stochastic, we provide standard deviations of these performance values across 10 realizations.

5.2.1 Spatial super-resolution

Here, one aims to reconstruct a high resolution image from a low resolution (i.e. downsampled) image. Downsampling is typically performed after lowpass filtering, and the downsampling factor and filter kernel determine the measurement model, M. Here, we use a 4×4 constant filter, and 4×4 downsampling (i.e., measurements are averages over non-overlapping blocks). We compare to two recent unsupervised methods Deep image Prior (DIP) (Ulyanov et al., 2020) and DeepRED (Mataev et al., 2019). DIP chooses a random input vector, and adjusts the weights of a CNN to minimize the mean square error between the output and the corrupted image. Regularization by denoising (RED) is a recent successful method closely related to P&P (Romano et al., 2017b). DeepRED (Mataev et al., 2019) combines DIP and RED, obtaining better performance than either method alone.

Inspection of results on two example images demonstrates that our method produces results

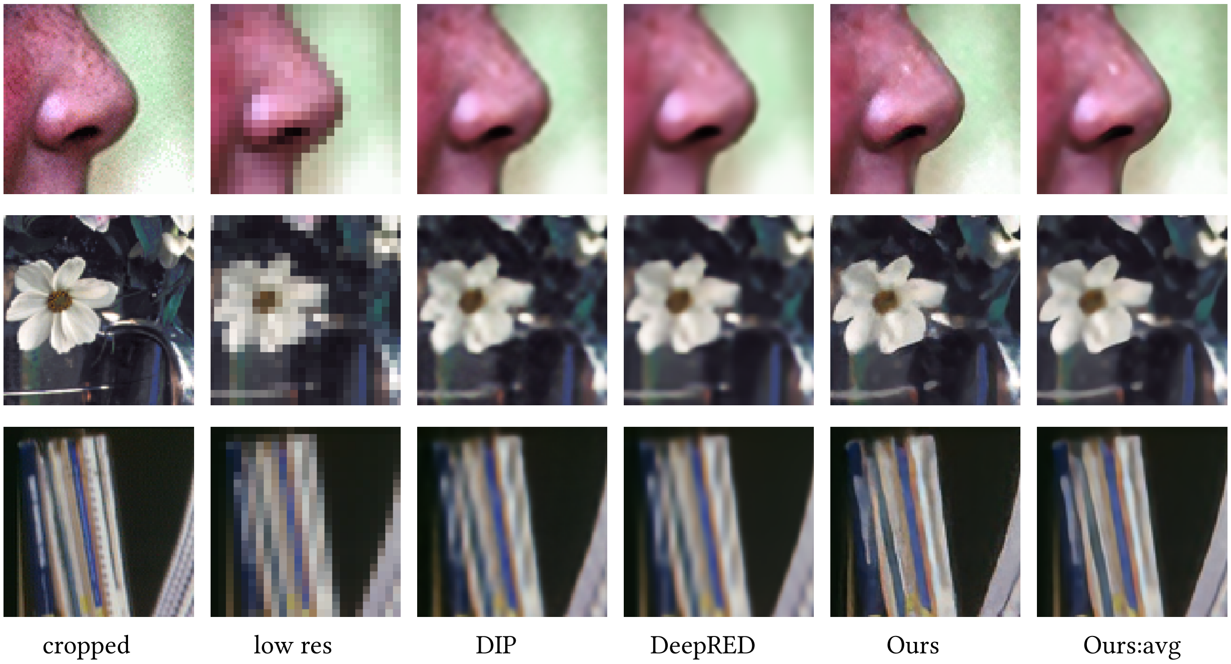


Figure 5.3: Spatial super-resolution. First column shows cropped portion of original images from Set14 (face: 272 × 272 and, Barbara: 720 × 576). The SISA algorithm can be applied to images of any resolution - we show cropped portions to facilitate visual inspection. Second column shows cropped portion with resolution reduced by averaging over 4x4 blocks (dimensionality reduction to 6.25%). Next three columns show reconstruction results obtained using DIP (Ulyanov et al., 2020), DeepRED (Mataev et al., 2019), and our method. In all cases, our method produces an image that is sharper with less noticeable artifacts (e.g., note blocking/aliasing artifacts along diagonal contours of lower image). The last column shows an average over 10 samples obtained by our method.

that are sharper with less noticeable artifacts (Fig. 5.3). Despite this, the PSNR and SSIM values are slightly worse than both DIP and DeepRED (Table 5.1). These can be improved by averaging over realizations (last column of Table 5.1), producing superior PSNR and SSIM values at the expense of some blurring (last column of Fig. 5.3). This is expected: the algorithm produces high-probability samples of the prior subject to the measurement constraint, but the least-squares optimal solution is the mean of the posterior distribution. If the samples are drawn from a curved manifold, their average (a convex combination of those points) will lie off the manifold. Finally, note that our method is more than two orders of magnitude faster than either DIP or DeepRED (bottom row, Table 5.1).

Table 5.1: Spatial super-resolution performance over Set5 (top 2 rows) and Set14 (second 2 rows). Values indicate YCbCr-PSNR (SSIM). Last row shows average Set14 runtime on a DGX GPU.

	MM^Tx	DIP	DeepRED	Ours±std	Ours:avg
4:1	26.35 (0.826)	30.04 (0.902)	30.22 (0.904)	29.47±0.09 (0.894±0.001)	**31.20** (**0.913**)
8:1	23.02 (0.673)	24.98 (0.760)	24.95 (0.760)	25.07±0.13 (0.767±0.003)	**25.64** (**0.792**)
4:1	24.65 (0.765)	26.88 (0.815)	27.01 (0.817)	26.56±0.09 (0.808±0.001)	**27.14** (**0.826**)
8:1	22.06 (0.628)	23.33 (0.685)	23.34 (0.685)	23.32±0.11 (0.681±0.002)	**23.78** (**0.703**)
runtime (sec):		1,190	1,584	**9**	10 × 9

Table 5.2: Spectral super-resolution (deblurring) performance over Set5, in YCbCr-PSNR (SSIM).

Ratio	MM^Tx	DIP	DeepRED	Ours±std	Ours:avg
10%	30.2 (0.91)	32.54 (0.93)	32.63 (0.93)	31.82±0.08 (0.93±0.001)	**32.78** (**0.94**)
5%	27.77 (0.85)	29.88 (0.89)	29.91 (0.89)	29.22±0.14 (0.89±0.002)	**30.07** (**0.90**)

5.2.2 Deblurring (spectral super-resolution)

The applications described above are based on partial measurements in the pixel domain. Here, we consider a blurring operator that retains a set of low-frequency coefficient in the Fourier domain, discarding the rest. This is equivalent to convolving the image with a sinc kernel. In this case, M consists of the preserved low-frequency columns of the discrete Fourier transform, and MM^Tx is the blurred version of x. Example images are shown in Fig. 5.4 and numerical comparisons are shown in Table 5.2. Our method produces strong results, both perceptually and numerically.

Table 5.3: Compressive sensing performance over Set68 (Martin et al., 2001). Values indicate PSNR (SSIM). TVAL3 (Li et al., 2013), ISTA-Net (Zhang and Ghanem, 2018), DIP (Ulyanov et al., 2020), BNN (Pang et al., 2020)

Ratio	TVAL3	ISTA-Net	DIP	BNN	Ours±std	Ours:avg
25%	26.48 (0.77)	29.07 (0.84)	27.78 (0.80)	28.63 (0.84)	29.16±0.033 (0.88±0.001)	**29.74(0.89)**
10%	22.49 (0.58)	25.23 (0.69)	24.82 (0.69)	25.24 (0.71)	25.47±0.03 (0.78±0.001)	**25.84 (0.80)**
4%	19.10 (0.42)	22.02 (0.54)	22.51 (0.58)	**22.52** (0.58)	22.07±0.05 (0.68±0.002)	22.29 (**0.69**)
runtime (sec)	3.1 (CPU)	0.04	?	1,680	32	32 × 10

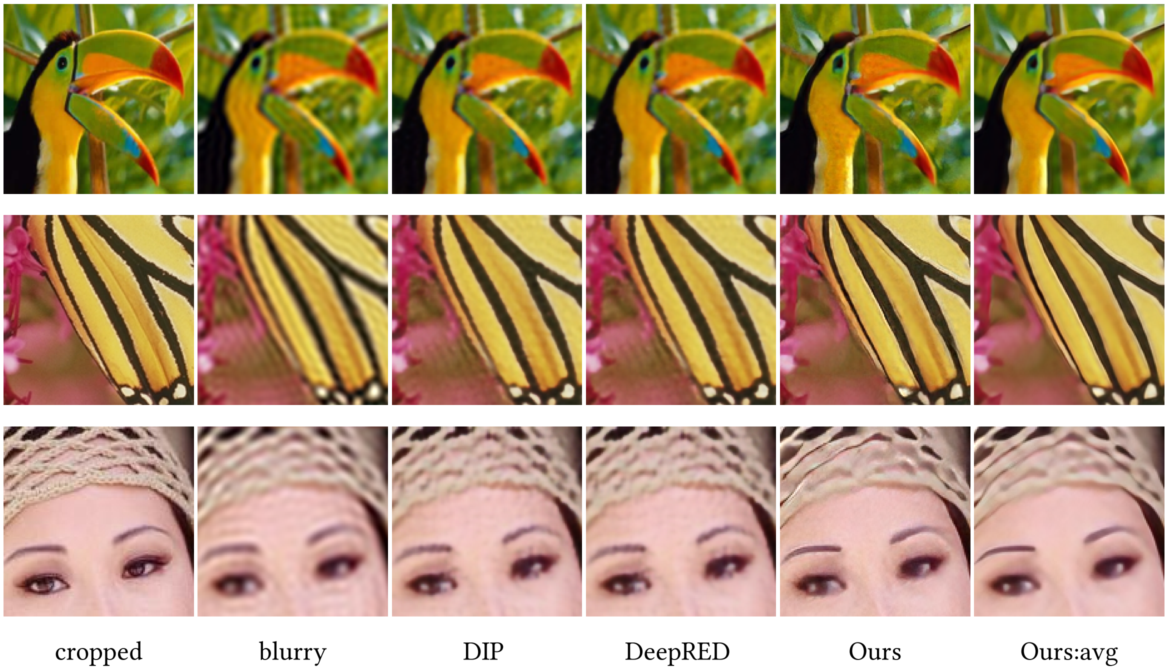


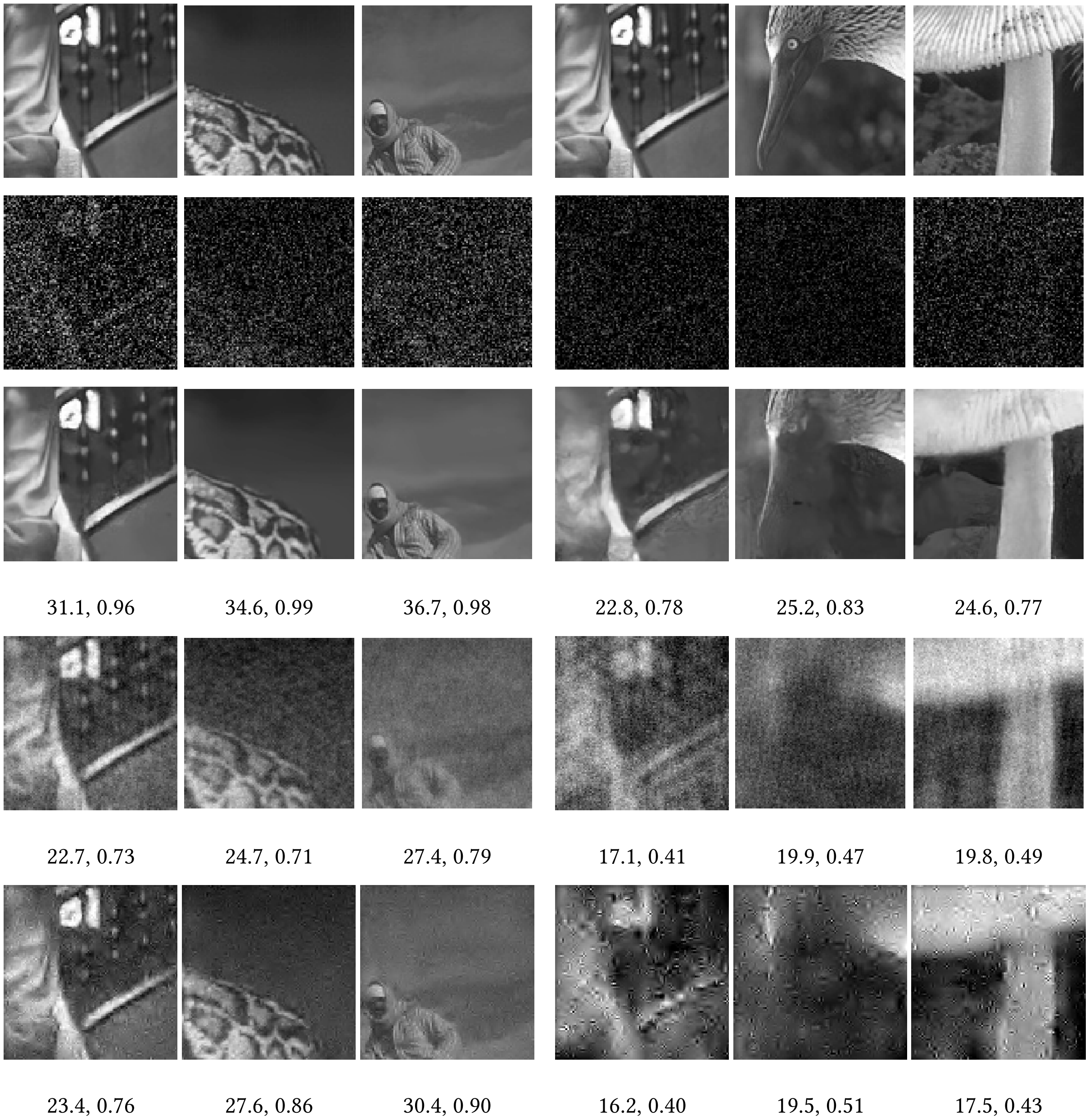
Figure 5.4: Deblurring (spectral super-resolution). Measurements correspond to lowest 5% of frequencies. Original images are from Set5 (butterfly of size 256 × 256 and woman of size 224 × 336.

5.2.3 Compressive sensing

Compressive sensing (Candès et al., 2006; Donoho, 2006) provides a set of theoretical results regarding recovery of sparse signals from a small number of linear measurements. Specifically, if one assumes that signals can be represented with at most $k << N$ non-zero coefficients in a known basis, they can be recovered from a measurements obtained by projecting onto a surprisingly small number of axes (approaching $k \log(N/k)$), far fewer than expected from traditional Shannon-Whitaker sampling theory. The theory relies on the sparsity property, which corresponds to a "union of subspaces" prior (Blumensath and Davies, 2009), and on the measurement axes being incoherent (essentially, weakly correlated) with the sparse basis. Typically, one chooses a sensing matrix containing a set of $n << N$ random orthogonal axes. Recovery is achieved by solving the sparse inverse problem, using any of a number of methods.

Photographic images are not truly sparse in any fixed linear basis, but they can be reasonably approximated by low-dimensional subsets of Fourier or wavelet basis functions, and compressive sensing results are typically demonstrated using one of these. The manifold prior embedded within our CNN denoiser corresponds to a nonlinear form of sparsity, and analogous to sparse inverse algorithms used in compressed sensing, SISA algorithm can be used to recover an image from a set of linear projections onto a set of random basis functions. Figure 5.5 shows four examples of images recovered from random projections using our denoiser-induced manifold prior, versus a sparse discrete cosine transform (DCT) prior. In all cases, the denoiser-recovered images exhibit sharper edges, more detail, and fewer artifacts. Numerical performance, in terms of both PSNR and SSIM, is consistent with the perceptual comparison.

We also compare to four other methods. TVAL3 (Li et al., 2013) is an optimization algorithm using total variation regularization, ISTA-Net (Zhang and Ghanem, 2018) is a block-based supervised CNN method trained to reconstruct images from measurements obtained from a single pre-specified measurement matrix. BNN (Pang et al., 2020) is an unsupervised Bayesian method for solving compressive sensing problems. DIP (Ulyanov et al., 2020) was previously described. Fig. 5.6 shows results for two example images, and Table 5.3 summarizes numerical performance. All values are taken from (Pang et al., 2020) except for ISTA-Net which were obtained by running the open-source code. Our method generally outperforms all other methods, even those that are specialized for Compressive Sensing (ISTA-NET, BNN).



31.1, 0.96 34.6, 0.99 36.7, 0.98 22.8, 0.78 25.2, 0.83 24.6, 0.77

22.7, 0.73 24.7, 0.71 27.4, 0.79 17.1, 0.41 19.9, 0.47 19.8, 0.49

23.4, 0.76 27.6, 0.86 30.4, 0.90 16.2, 0.40 19.5, 0.51 17.5, 0.43

Figure 5.5: Compressive sensing. Measurement matrix M contains random, orthogonal unit vectors, with dimensionality reduced to 30% for the first three columns, and 10% for last three columns. As in previous figures, top row shows original images (x), and second row is linear (pseudo-inverse) reconstruction ($MM^T x$). Third row: images recovered using our method. Fourth row: standard compressive sensing solutions, assuming a sparse DCT signal model. Last row: standard compressive sensing solutions, assuming a sparse wavelet (db3) signal model. Numbers indicate performance, in terms of PSNR (dB), and SSIM (Wang et al., 2004b).

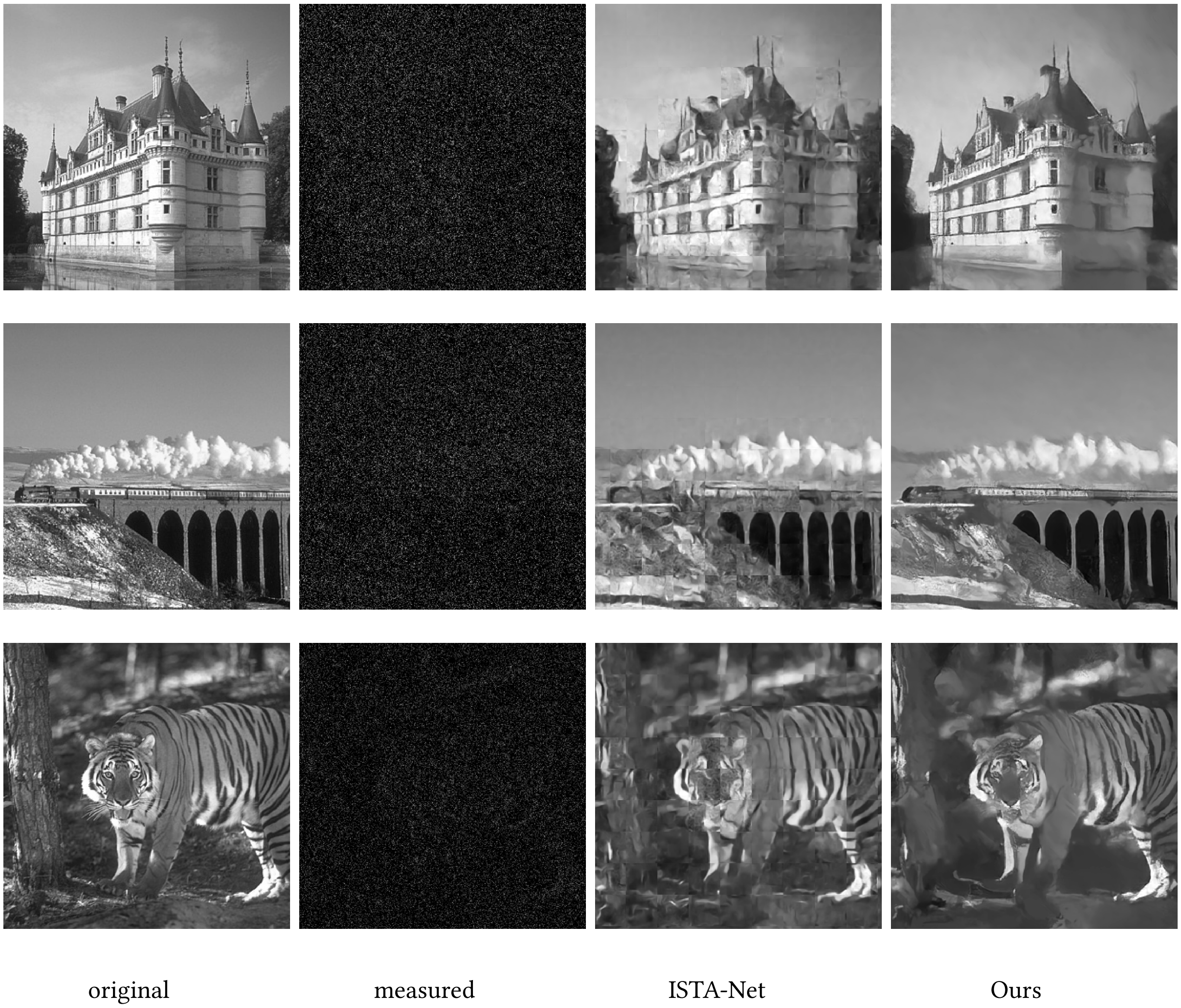


Figure 5.6: comparison with a deep net method in compressive sensing. Measurement matrix M contains random orthogonal unit vectors, whose number is equal to 10% of the number of image pixels (300×300). Third column: images recovered using ISTA-Net (Zhang and Ghanem, 2018)(trained with supervision for a particular measurement matrix). Fourth column: Our method, which again is seen to exhibit fewer artifacts.

5.2.4 Inpainting

A simple example of a linear inverse problem involves restoring a block of missing pixels, conditioned on the surrounding content. Here, the columns of the measurement matrix M are a subset of the identity matrix, corresponding to the measured (outer) pixel locations. We choose a missing block of size 30×30 pixels, which is less than the size of the receptive field of the BF-CNN network (40×40), the largest extent over which this denoiser can be expected to directly capture joint statistical relationships. Figure 5.7 shows examples. Figure 5.8 illustrates the stochasticity of our method in solving inverse problems.

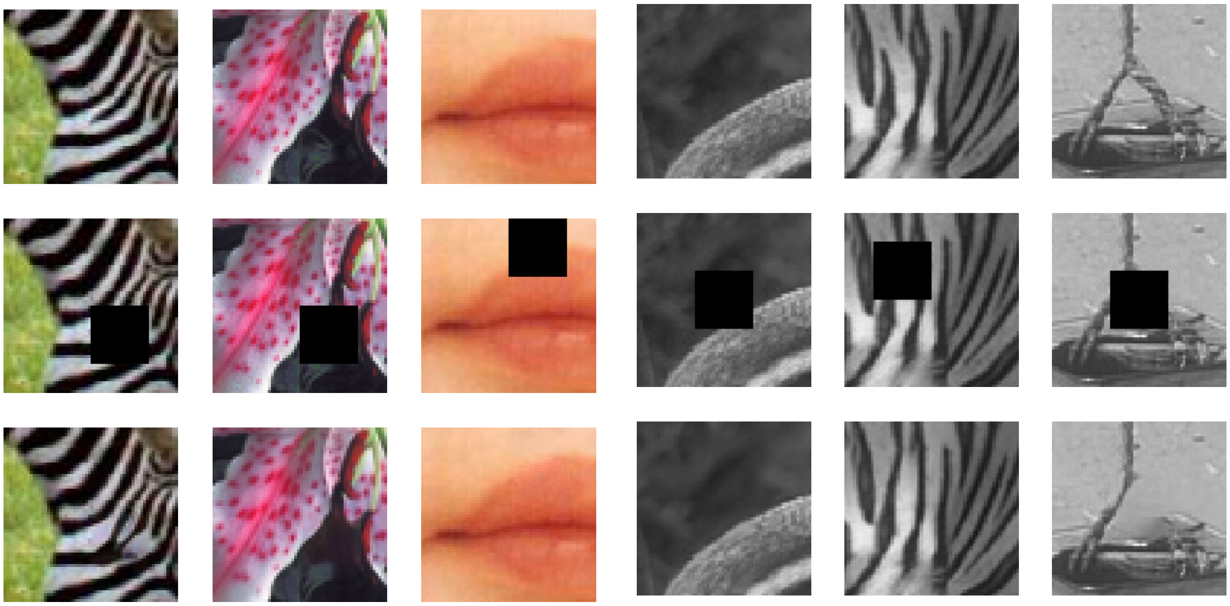


Figure 5.7: Inpainting examples. Top row: original images (x). Middle: Images corrupted with blanked region ($MM^T x$). Bottom: Images restored using SISA algorithm.

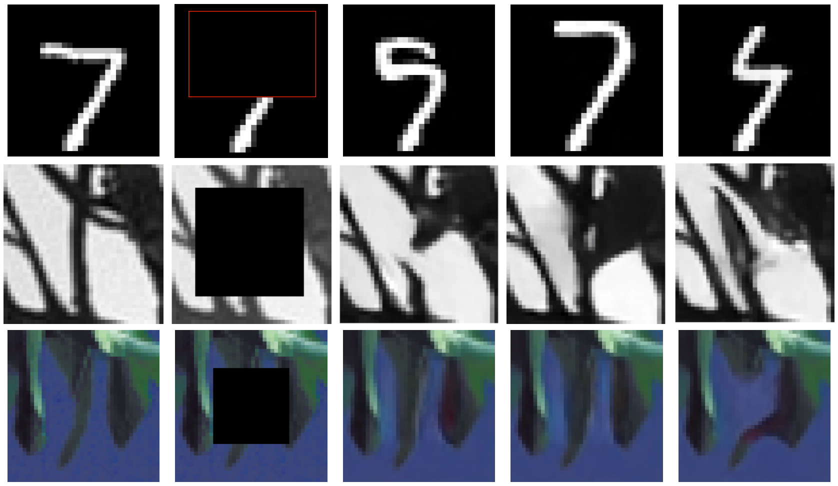


Figure 5.8: Sampling diversity. Inpainting examples generated using two BF-CNN denoisers trained on (1) MNIST, (2) Berkeley grayscale segmentation dataset (3) Berkeley color segmentation dataset. First column: original images. Second column: partially measured images, with missing block . Right three columns: Restored examples, arising from different random initializations, y_0.

5.2.5 Random missing pixels

Consider a measurement process that discards a random subset of pixels. M is a low rank matrix whose columns consist of a subset of the identity matrix corresponding to the randomly chosen set of preserved pixels. Figure 5.9 shows examples with 10% of pixels retained. Despite the significant number of missing pixels, the recovered images are remarkably similar to the originals.

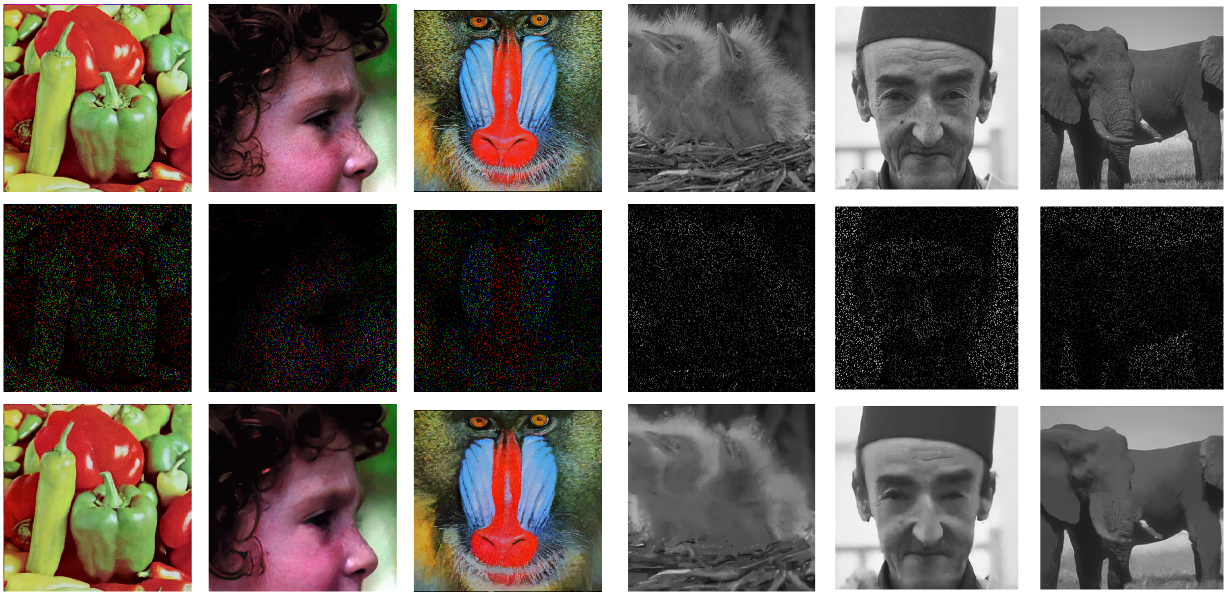


Figure 5.9: Recovery of randomly selected missing pixels. 10% of dimensions retained. Top row: original images, x. Middle: projection onto measurements, $MM^T x$, Bottom: images recovered using SISA algorithm.

5.2.6 COLORIZATION

Another example of a linear inverse problem is colorization. A grayscale image can be interpreted as a partial measurement of a color image, where the measurement is the sum of the three RGB channels. The measurement matrix can be constructed accordingly. Figure 5.10 shows examples of grayscale image, colorized using the universal inverse problem algorithm. The colors are realistic and show consistency across continuous region.

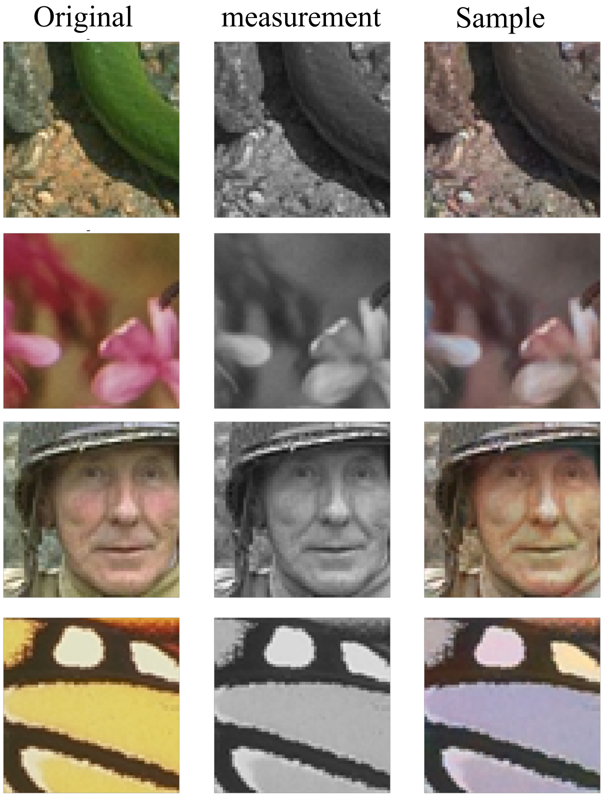


Figure 5.10: Colorization of grayscale images. Left column: original images, x. Middle: grayscale images. Right column: colorized images

5.2.7 Reconstruction from retinal cone excitations

We [2] can use the inverse problem solver to understand representations of biological vision. When photons land on the retina, they elicit responses from the cone photoreceptor. This response is a partial measurement of the totality of light reflected from an object. Our brains, arguably, rely on an internal prior of images to make sense of this partial measurement.

An interesting observation about human retina is its huge diversity in terms of proportions of the three types of cone cells. In most humans, cone cells in the retina consist of three types: L, M, S, which respond to long, medium and short wavelength respectively. Surprisingly, the ratio of

[2] This is part of the result presented in (Zhang et al., 2022b) in collaboration with Ling-Qi Zhang and David Brainard.

L-cones to M-cones varies tremendously across individuals (Hofer et al., 2005) (see top part of Figure 5.11). However, it appears that our perception is invariant across this diversity. It turns out that the denoiser prior is aligned with human prior in this regard.

We use the iSet-Bio package (Cottaris et al., 2019) to simulate the cone photoreceptor array of 8 individuals, ranging from low to high L-cone to M-cone ratio. The red asterisks in Figure 5.11 mark these ratios for human subjects. Then we use our method to reconstruct images from the simulated cone measurements. The plot on the right in Figure 5.11 shows the error of reconstruction using the denoiser prior. Each curve shows contours of constant reconstruction error. For comparison, on the left we show the reconstruction error using a sparse image prior. Interestingly, all the human cones are similar in terms of reconstruction error and they all fall within the elongated global minimum area. On the contrary, the sparse prior leads to different reconstruction error in different subjects, which is not consistent with perceptual reports.

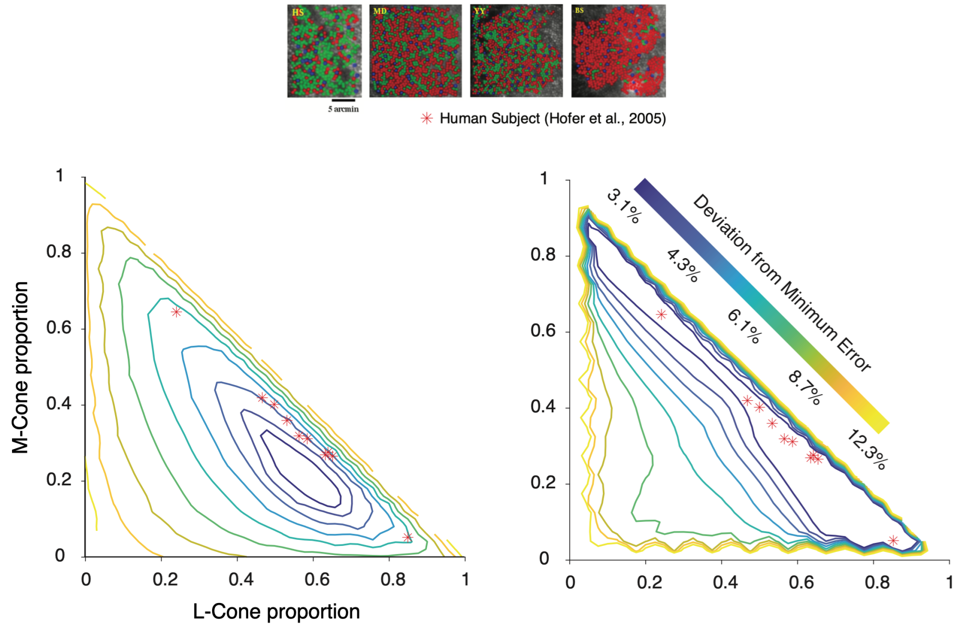


Figure 5.11: Reconstruction from cone excitations is robust to cone cell proportions under the denoiser prior. **Top.** Example data from 4 human subjects. **Bottom right.** Contour plot pf reconstruction error assuming denoiser prior. The measurements are taken from 8 simulated retina shown by asterisks. **Bottom left.** Contour plot pf reconstruction error assuming sparse prior (Zhang et al., 2022a).

5.2.8 DE-QUANTIZATION

Image compression is an important tool in signal communication. The most popular compression technique is JPEG, in which the image is encoded using block DCT. Then coefficients are quantized into bins. The severity of quantization varies depending on the desired size of the compressed image, determined by a trade-off between the quality and the number of bits (rate-distortion) allocated. Decoding the image, specially from severely compressed images causes visible distortions in the form of blur, artifact and blocking effects. We use Algorithm 3 to remove these artifacts and improve the quality of the image using the denoiser prior.

This problem is a simple case of a non-linear inverse problem, in which the measurements (quantized DCT coefficients) are constrained to be within a convex set (the corresponding bin), rather than lying on a hyper-plane. We modify the algorithm to handle projection onto a convex set. At each iteration, the image is modified following the direction of gradient provided by the denoiser. Then the constraint is enforced by projecting the altered coefficients to the convex set determined by the bin boundaries.

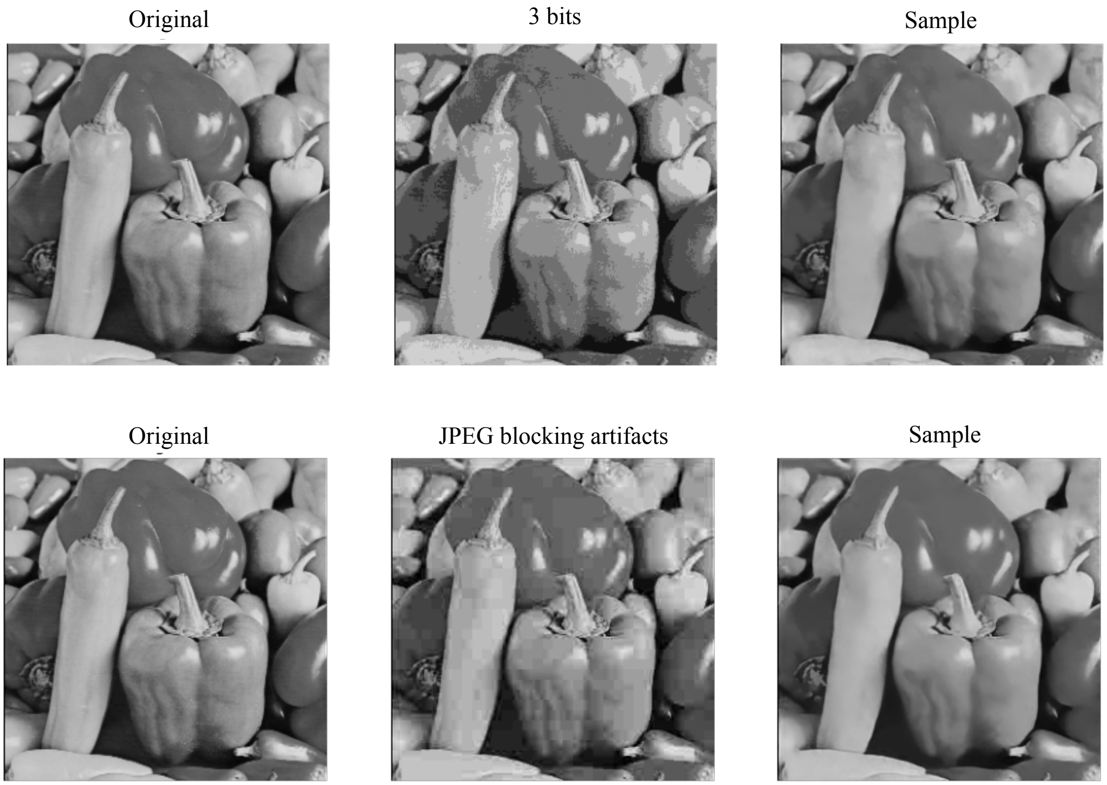


Figure 5.12: de-quantization. Top row: An original 8-bit image is quantized in the pixel domain to a 3-bit image. The last column shows a reconstruction from the 3-bit image. Bottom row: The same image is compressed using JPEG method, and improved through reconstruction using the deoniser prior.

5.2.9 RECONSTRUCTING FROM NOISY MEASUREMENTS

Finally, we extend our algorithm to handle noisy partial measurements. In real world, all measurements obtained by sensing systems are noisy. Here we assume Gaussian noise for simplicity. Generalizing this result to other noise models is a future direction, with important implications for practice.

Intuitively, when the measurements are noisy, we do not aim to perfectly satisfy the constraint, as that results in noisy reconstruction. With increasing measurement noise, the reliance on the measurement decreases and the reconstruction becomes more prior-reliant. Hence, we modify the update line in Algorithm 3 to account for the measurement noise:

$$d_t = (I - \frac{\sigma^2}{\sigma^2 + \sigma_w^2} M M^T) f(y_{t-1}) + \frac{\sigma^2}{\sigma^2 + \sigma_w^2} M(x_c - M^T y_{t-1})$$

where σ_w is the standard deviation of the measurement noise.

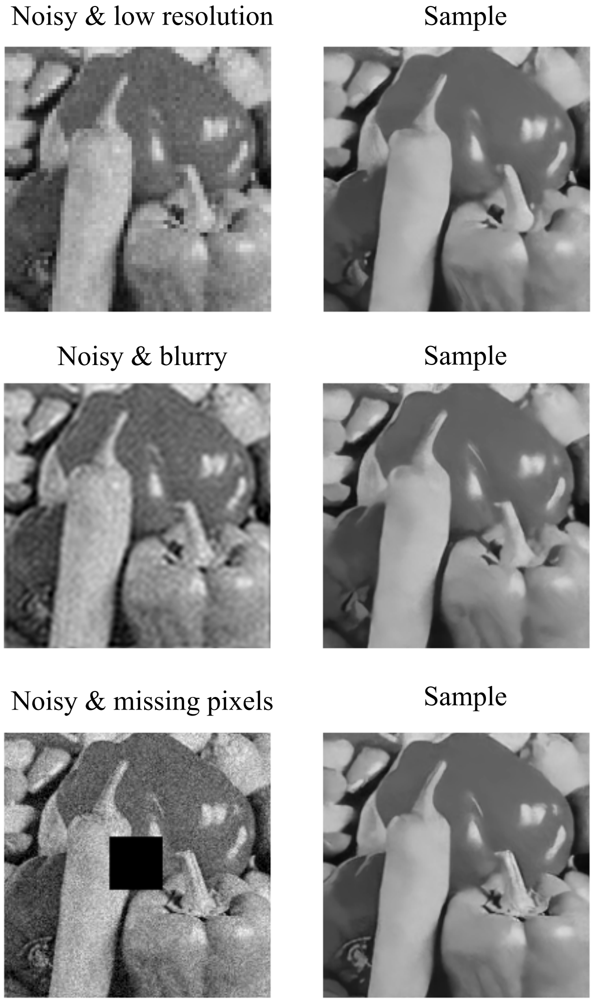


Figure 5.13: reconstructing from noisy measurements

Table 5.4: Run time, in seconds, and (number of iterations until convergence) for different applications, averaged over images in Set12, on an NVIDIA RTX 8000 GPU. The cost of running the algorithm can be divided into two parts: number of iterations, and cost per iteration. Number of iterations for all applications except for inpainting is in the same order. Solving inpainting with this algorithm requires information to spread in from the borders of the missing block, which in turn requires more iterations. Cost of each iteration, aside from the cost of denoiser's forward pass, comes from MM^Tx operation. For all applications, except for CS, dot products can be reduced to more efficient operations such as Hadamard products or filtering. In CS, M contains random vectors, so measurements require full matrix multiplication. Consequently, CS takes longer per iteration.

Inpainting 30 × 30	Missing pixels 10%	SISR 4:1	Deblurring 10%	CS 10%
15.7 (647)	7.8 (313)	7.6 (301)	8.4 (332)	32.5 (317)

5.3 Optimal linear measurements

In Section 5.2 we demonstrated that the denoiser prior can be utilized to solve *any* linear inverse problems. A striking observation regarding these solutions is that for a given image and a fixed number of measurements, the quality of the solution is highly dependent on the task itself. In other words, not all measurement procedures are equally effective if the objective is to reconstruct the image from a low dimensional measurement using the prior. This observation gives rise to a fundamental question: Given a specific signal prior, what is the *optimal* set of low-dimensional linear measurements?

The optimality of the measurements is defined with respect to the prior. For example, for a Gaussian prior, the optimal set of linear measurements are the top principal components (PCs), and the minimal squared error reconstruction is in turn achieved through a simple linear projection onto those PCs. However, the Gaussian prior falls short of capturing important higher-order dependencies in natural images, resulting in poor reconstruction quality.

With the emergence of sparse priors, an important breakthrough in developing better linear

Results in this section are presented in (Zhang et al., 2024) in collaboration with Ling-Qi Zhang and David Brainard measurements arose. The seminal work by Donoho (Donoho, 2006) in compressed sensing (CS) proved that when the signal lies within a union of subspaces (a type of sparse prior), the optimal measurements are incoherent with the axes of the signal subspaces, and are well-approximated by a set of randomly chosen vectors. Under this setting, an iterative non-linear reconstruction from low-dimensional linear measurements can achieve near-perfect recovery (Tropp, 2006). The sparse prior is well-suited to describe some signal classes such as medical images, leading to significant empirical improvements.

The statistical characterization underlying PCA and CS assumes alignment of signal content along principal component axes or subspaces, but this assumption does not adequately capture the statistical structure of natural images (Ballé et al., 2016; Portilla et al., 2003b). In support of this, previous work (Weiss et al., 2007) has demonstrated that random measurements outperform PCs only on idealized sparse signals, but not on natural images.

We re-visit the question of optimal linear measurement for inverse problems, with respect to the prior implicit in a denoiser. Specifically, we develop a framework for optimizing a set of linear measurements in order to minimize the error obtained from nonlinear reconstruction via Algorithm 3. This enables us to analyze the impact of natural image statistics on the Optimized Linear Measurements (OLM). We demonstrate that these measurements

(1) vary substantially with the training dataset (e.g., digits vs. faces); See Figure 5.14

(2) vary with the choice of reconstruction loss (e.g., MSE vs. SSIM); See Figure 5.15

(3) are distinct from those of PCA and CS; See Figure 5.14

(4) lead to substantial performance improvements over PCA and CS. See Figure 5.16 and 5.17

This work provides yet another example of the impressive improvements that can be achieved by applying modern ML methods to foundational problems of signal processing. For more detail, see (Zhang et al., 2024).

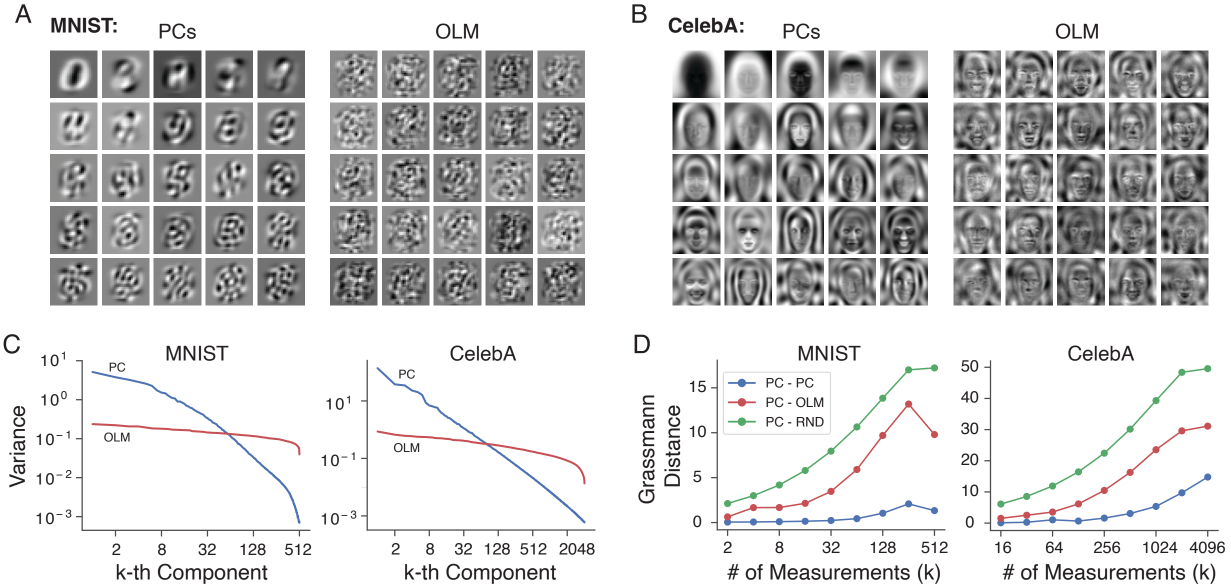


Figure 5.14: Analysis of optimized measurement vectors. **A)** Left: Twenty-five selected principal components of the MNIST dataset are visualized as images. They are ordered by variance explained from top to bottom, and left to right. Right: Twenty-five components from the columns of the OLM for $k = 64$. **B)** Same as **A)** but for the CelebA dataset. **C)** The variance of the measurement on each component as a function of number of measurements (k) for the PCs (blue) and OLMs (red). Note that the optimal measurements are sorted post hoc based on the variance explained by each component. These vectors are optimized jointly for each k and are not ordered by the algorithm itself. **D)** The Grassmann distance betweensubspaces spanned by different linear measurements as a function of k (Hamm and Lee, 2008). We show the distance between subspaces spanned by PCs obtained on two random halves of the training data (blue), between subspaces spanned by PCs and OLM (blue), and between subspaces spanned by PCs and a set of random measurements (green).

5.4 Related work

Nearly a decade ago, a strategy known as Plug-and-Play (P&P) was proposed for using a denoiser as a regularizer in solving other inverse problems (Venkatakrishnan et al., 2013), and a number of recent extensions have used this concept to develop MAP solutions for linear inverse problems (Chan et al., 2017; Kamilov et al., 2017; Mataev et al., 2019; Meinhardt et al., 2017; Reehorst and Schniter, 2019; Romano et al., 2017b; Sun et al., 2019; Teodoro et al., 2019; Zhang et al., 2017c). Generally, the objective is decoupled into data fidelity and regularization terms, introducing a slack variable for use in a proximal optimization algorithm (e.g., ADMM). The proximal operator of

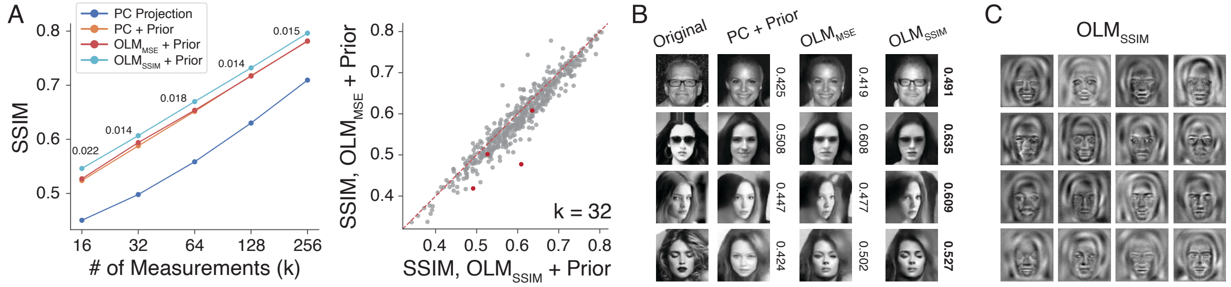


Figure 5.15: Optimal measurement for perceptual loss (structural similarity index measure - SSIM) (Wang et al., 2004a). **A)** Left: SSIM as a function of number of measurement k, for PC projection, PC, and OLM optimized for MSE and SSIM, all three using the denoiser prior. The number indicates the increase in SSIM from OLM optimized for MSE to OLM optimized for SSIM. Right: Scatter plot of the SSIM value for individual images, for OLM optimized for SSIM (x-axis) and MSE (y-axis), respectively. **B)** Example linear inverse solutions obtained from different measurement matrices combined with the denoiser prior. The numbers indicate the SSIM value for each individual image. **C)** Example measurement vectors from the OLM optimized for SSIM.

the regularization term is interpreted as the MAP solution of a denoising problem, and is replaced by a denoiser. The main drawback of the P&P framework is its reliance on a MAP denoiser, which is not easy to obtain. As a result, virtually all methods within this framework replace the MAP denoiser with an MMSE denoiser in practice, which breaks the convergence rules. However, recent publications have proven convergence of such algorithms when used in conjunction with MMSE denoisers (Laumont et al., 2021; Xu et al., 2020).

Our method is conceptually similar to P&P in that it uses a denoiser to solve linear inverse problems. But it differs in a number of important ways: (1) *direct relationship between prior and denoiser mapping.* The rationale for using a denoiser as a regularizer in P&P frameworks arises from interpreting the proximal operator of the regularizer as a MAP solution of a denoising problem, providing an indirect connection to prior; We derive our method from Miyasawa's relationship between an MMSE denoiser and the prior, which is exact and explicit, and makes the algorithm interpretable. This connection has also been made in (Laumont et al., 2021; Xu et al., 2020). (2) *sampling vs. MAP estimation.* We obtain a high probability image from the implicit prior that is consistent with a linear constraint. Our solution is stochastic, and does not minimize MSE,

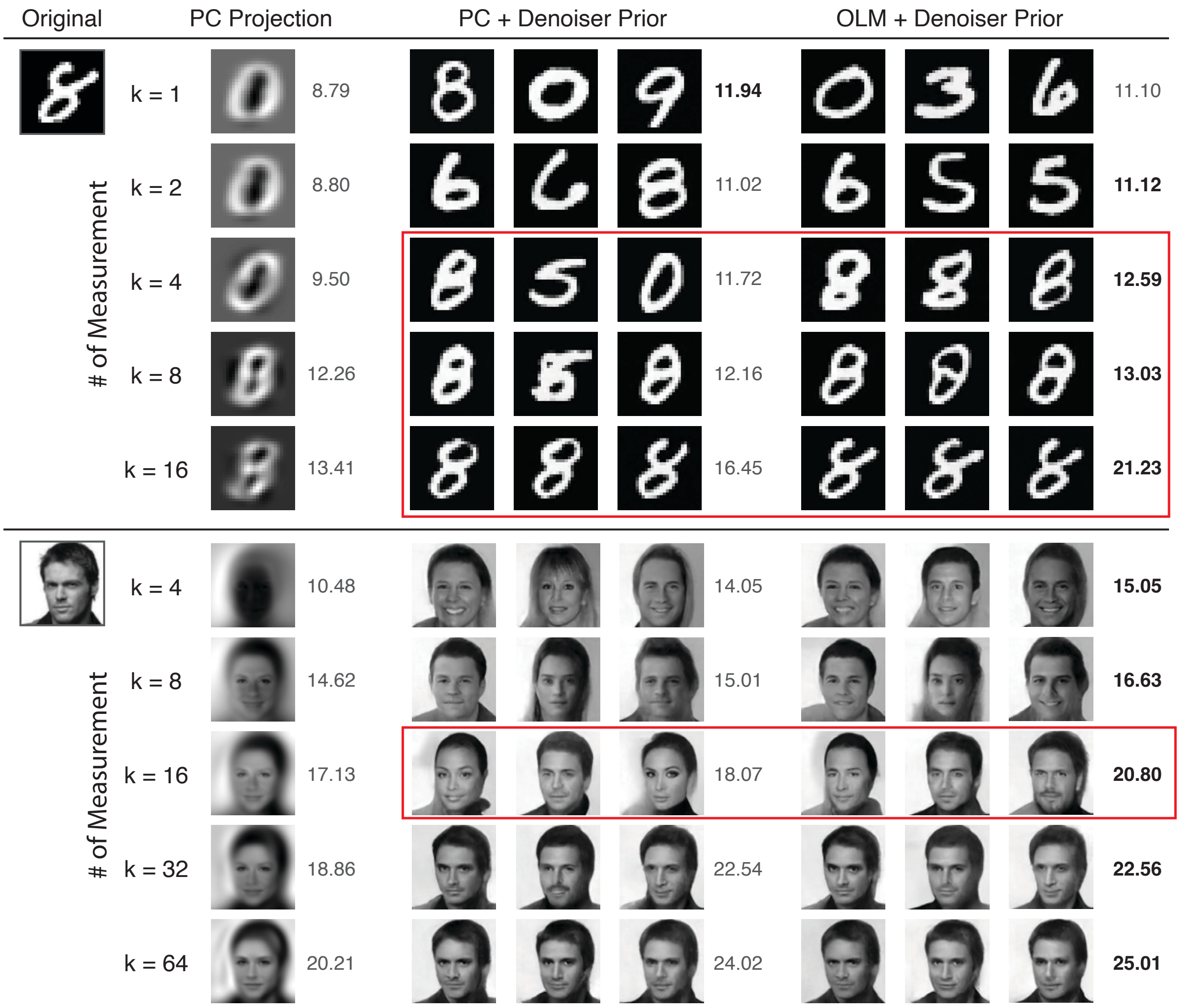


Figure 5.16: Reconstruction from optimal measurements. Columns show the PC projection, three conditional samples from the denoiser prior using the PCs, and three samples using the OLMs, respectively. Rows correspond to increasing number of measurements, k. The numbers indicate the PSNR value of the reconstructions, obtained by averaging over 16 samples.

but has high perceptual quality. RED and other P&P methods are derived as MAP solutions, and although this is not equivalent to minimizing MSE (maximizing PSNR), the results generally have better PSNR than our sampling results, but are visually more blurred (see Figs. 5.3 and 5.4); (3) *automatic step-size selection and convergence.* P&P, which uses ADMM for optimization, relies on hyper-parameter selection that are critical for convergence (as discussed extensively in (Romano et al., 2017b)). Our algorithm adjusts step-size automatically using the blind denoiser, with only

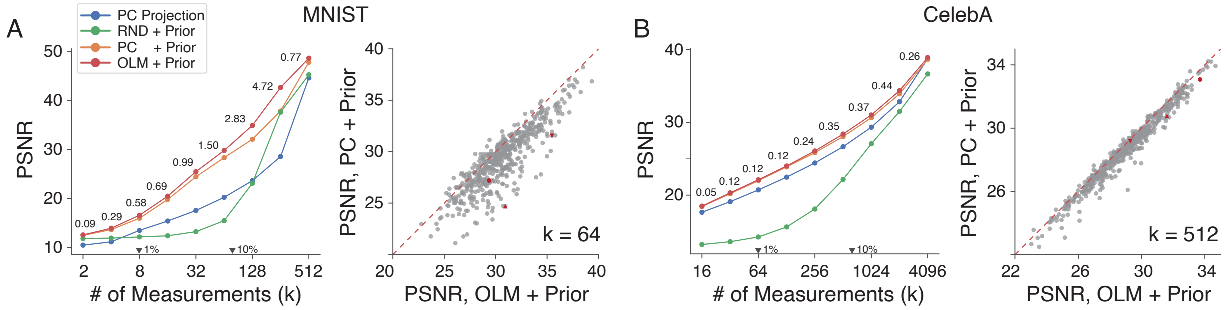


Figure 5.17: Performance of the optimized linear measurements, for **A)** MNIST and **B)** CelebA datasets. Left panel shows the peak signal-to-noise ratio (PSNR) as a function of the number of measurements k, for linear reconstruction from PCs (blue), denoiser prior reconstruction from PCs (orange), denoiser prior reconstruction from random measurements (green), and denoiser prior reconstruction from OLMs (red). Numbers indicate increase in PSNR from PC + denoiser prior compare to OLM + prior. The right panel shows a scatter plot comparing the PSNR values using the OLM (x-axis), and PCs (y-axis), over the images in the test set, for a value of k around 10% of the total number of pixels. Red points mark those shown in the examples of Figure 5.16.

two primary hyper-parameters (h_0 and β), and convergence is robust to choices of these (Fig. 2.2).

5.5 Discussion

We've described a framework for using the prior embedded in a denoiser to solve inverse problems. Specifically, we developed a stochastic iterative score ascent algorithm that uses the denoiser to draw high-probability samples from its implicit prior, and a constrained variant that can be used to solve *any* deterministic linear inverse problem. To demonstrate the generality of our algorithm, we showed solving several inverse problems using the same algorithm and parameters, without any additional training. Finally, we empirically demonstrated its efficiency in Table 5.1. Denoisers can be trained supervised on unlimited amounts of unlabeled data, and as such, our method extends the power of supervised learning to a much broader set of problems, with no additional training.

As mentioned previously, the performance of our method is not well-captured by comparisons to the original image (using, for example, PSNR or SSIM). Performance should ultimately be tested using experiments with human observers, which might be approximated using a no-reference perceptual quality metric (e.g., (Ma et al., 2018)). Handling of nonlinear inverse problems is a natural extension of the method which requires modification of the algorithm. Another potential extension is generalization to stochastic linear inverse problems, $x^c = Mx + e$, where e is any type of noise.

Additionally, we showed that we can find optimal linear measurements, using our reconstruction method based on a learned prior embedded in a denoiser. We showed numerically that the set of linear measurements found through our method, result in superior image reconstruction. This result shows that for signals with non-Gaussian distributions, better measurements exist for minimizing MSE, even though projections onto the PCs maximize the explained variance. We show that the optimal measurement are sensitive to both the statistics of the image dataset, and to the objective function for which they are optimized. Our results highlight the importance of accurately modeling the statistics of the signals to design efficient linear measurements.

6 | Discussion

This thesis explores image density models in the era of deep learning. We propose a methodology to learn a prior from data and sample from it, which was thought to be impossible before the advent of deep neural networks. The learned prior is sophisticated, powerful, and mysterious, hence a very interesting subject of study.

A foundational question regarding the learned prior is whether it is a "good" approximation of the true underlying density. Indeed measuring the optimality of the density is not straightforward since we do not (and will never) have access to the true image density. Regardless, we can invent and use proxies for evaluating optimality. We approach this question in Chapter 3 by designing an experiment to measure the convergence of model variance. We showed that, with large yet realizable datasets, the learned density becomes independent of the specific examples in the training set, indicating strong generalization defined as zero model variance. In this regime the learned prior is well-defined over a continuous set, achieved through interpolation between training images. Thereby, samples drawn from a density model in the generalization regime are novel and diverse. On the contrary, if the model manages to memorize the training examples, the diversity of the samples is limited to the diversity of the training set. This case corresponds to a high model variance, since changing the training set maximally affects the learned density.

A low model variance, or high sample diversity, is not a sufficient measure of a "good" density approximation. In addition to high sample diversity, we would like high sample quality. Zero model variance can be achieved by reducing the neural network capacity, which can result in diverse but low quality samples. For a model with zero variance, high sample quality is a proxy for low model bias. Measuring sample quality, although difficult, is more tractable than evaluating model bias.

In Chapter 3 we demonstrated that for a fixed image resolution and network capacity, the model resides in three different regimes as a function of training set size. For small number of training images, the network memorizes the training set, leading to high quality but low diversity samples. By increasing the dataset, the model enters a transition phase where the diversity increases but the quality degrades drastically. Eventually the model enters a generalization regime, where both image quality and diversity are high. This feat is due to the alignment of inductive biases of the network with the characteristics of the distribution. These inductive biases manifest themselves in the form of an adaptive harmonic basis. Assuming a manifold hypothesis, top directions of the basis reveal the tangent plane of the manifold locally.

Revealing the inductive biases of the network is key in understanding the prior and improving it. To this end, in Chapter 2 we modify the architecture to achieve algebraic homogeneity, which promotes intensity invariance in the learned prior. Along the same lines, in Chapter 4, we introduce a conditional local multi-scale architecture to verify and incorporate a locality assumption about images. The results presented in the chapter demonstrate that introducing inductive biases inspired by image distribution structures directly results in reducing network parameters, hence dramatically decreasing the number of training images needed to generalize.

Having access to learned priors embedded in DNN naturally leads to many more questions and opens numerous research possibilities. An important component missing from the picture is an explicit internal representation. Unlike VAEs, it is not immediately obvious where in the network the internal representation reside and how to access it. If successfully obtained, then it can be used as an effective embedding in a variety of downstream tasks, such as solving non-linear inverse problems, proper conditioning during sampling, or defining similarity metrics.

This thesis present a collection of studies that fall under the emerging field of *science of* *deep learning*. We used carefully designed experiments to validate or falsify hypotheses formed through empirical observations and intuitions. The ultimate goal is not to beat the state-of-the-art performance, but to *understand* the mechanisms of deep nets and then employ the understanding to improve *generalization* properties of our models. And we have just begun!

A | Learning image densities from data using a denoiser

A.1 Description of denoising architectures

In this section we describe the denoising architectures used for our computational experiments in more detail.

A.1.1 DnCNN

We implement BF-DnCNN based on the architecture of the Denoising CNN (DnCNN) (Zhang et al., 2017a). DnCNN consists of 20 convolutional layers, each consisting of 3×3 filters and 64 channels, batch normalization (Ioffe and Szegedy, 2015), and a ReLU nonlinearity. It has a skip connection from the initial layer to the final layer, which has no nonlinear units. To construct a bias-free DnCNN (BF-DnCNN) we remove all sources of additive bias, including the mean parameter of the batch-normalization in every layer (note however that the scaling parameter is preserved).

A.1.2 Recurrent CNN

Inspired by (Zhang et al., 2018a), we consider a recurrent framework that produces a denoised image estimate of the form $\hat{x}_t = f(\hat{x}_{t-1}, y_{\text{noisy}})$, at time t where f is a neural network. We use a 5-layer fully convolutional network with 3×3 filters in all layers and 64 channels in each intermediate layer to implement f. We initialize the denoised estimate as the noisy image, i.e $\hat{x}_0 := y_{\text{noisy}}$. For the version of the network with net bias, we add trainable additive constants to every filter in all but the last layer. During training, we run the recurrence for a maximum of T times, sampling T uniformly at random from $\{1, 2, 3, 4\}$ for each mini-batch. At test time we fix $T = 4$.

A.1.3 UNet

Our UNet model (Ronneberger et al., 2015a) has the following layers:

1. *conv1* - Takes in input image and maps to 32 channels with 5×5 convolutional kernels.
2. *conv2* - Input: 32 channels. Output: 32 channels. 3×3 convolutional kernels.
3. *conv3* - Input: 32 channels. Output: 64 channels. 3×3 convolutional kernels with stride 2.
4. *conv4*- Input: 64 channels. Output: 64 channels. 3×3 convolutional kernels.
5. *conv5*- Input: 64 channels. Output: 64 channels. 3×3 convolutional kernels with dilation factor of 2.
6. *conv6*- Input: 64 channels. Output: 64 channels. 3×3 convolutional kernels with dilation factor of 4.
7. *conv7*- Transpose Convolution layer. Input: 64 channels. Output: 64 channels. 4×4 filters with stride 2.

8. *conv8*- Input: 96 channels. Output: 64 channels. 3 × 3 convolutional kernels. The input to this layer is the concatenation of the outputs of layer *conv7* and *conv2*.

9. *conv9*- Input: 32 channels. Output: 1 channels. 5 × 5 convolutional kernels.

The structure is the same as in (Zhang et al., 2018a), but without recurrence. For the version with bias, we add trainable additive constants to all the layers other than *conv9*. This configuration of UNet assumes even width and height, so we remove one row or column from images in with odd height or width.

A.1.4 Simplified DenseNet

Our simplified version of the DenseNet architecture (Huang et al., 2017) has 4 blocks in total. Each block is a fully convolutional 5-layer CNN with 3 × 3 filters and 64 channels in the intermediate layers with ReLU nonlinearity. The first three blocks have an output layer with 64 channels while the last block has an output layer with only one channel. The output of the i^{th} block is concatenated with the input noisy image and then fed to the $(i+1)^{th}$ block, so the last three blocks have 65 input channels. In the version of the network with bias, we add trainable additive parameters to all the layers except for the last layer in the final block.

A.2 Datasets and training procedure

Our experiments are carried out on 180 × 180 natural images from the Berkeley Segmentation Dataset (Martin et al., 2001). We use a training set of 400 images. The training set is augmented via downsampling, random flips, and random rotations of patches in these images (Zhang et al., 2017a). A test set containing 68 images is used for evaluation. We train the DnCNN and it's bias free model on patches of size 50 × 50, which yields a total of 541,600 clean training patches. For the remaining architectures, we use patches of size 128 × 128 for a total of 22,400 training patches.

We train DnCNN and its bias-free counterpart using the Adam Optimizer (Kingma and Ba, 2014) over 70 epochs with an initial learning rate of 10^{-3} and a decay factor of 0.5 at the 50^{th} and 60^{th} epochs, with no early stopping. We train the other models using the Adam optimizer with an initial learning rate of 10^{-3} and train for 50 epochs with a learning rate schedule which decreases by a factor of 0.25 if the validation PSNR decreases from one epoch to the next. We use early stopping and select the model with the best validation PSNR.

A.3 Network bias impairs generalization across architectures

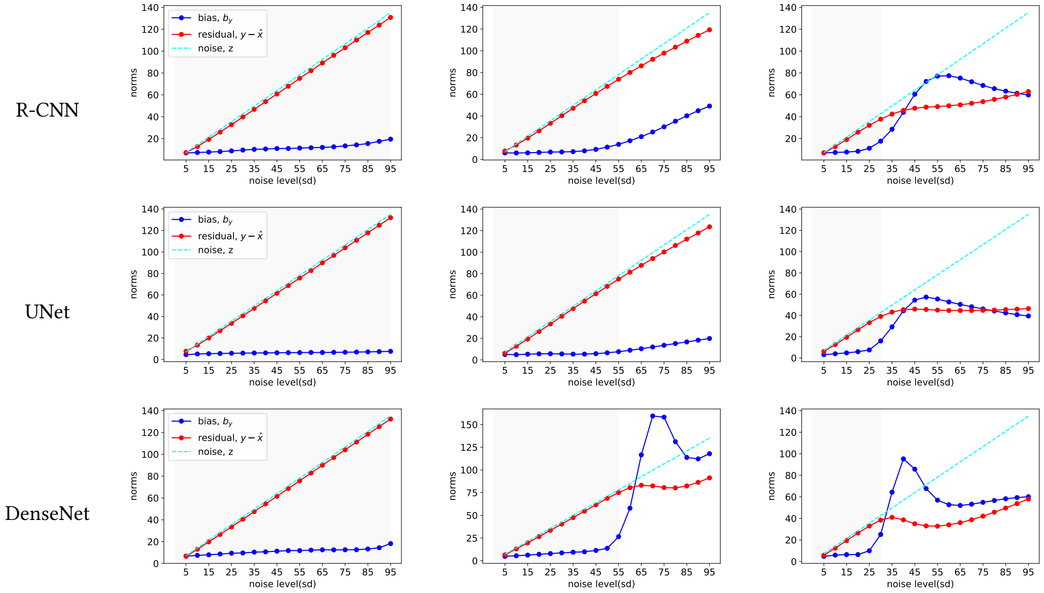
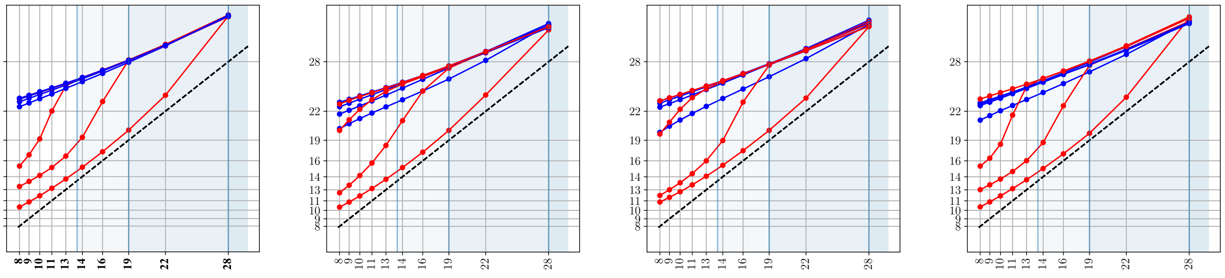
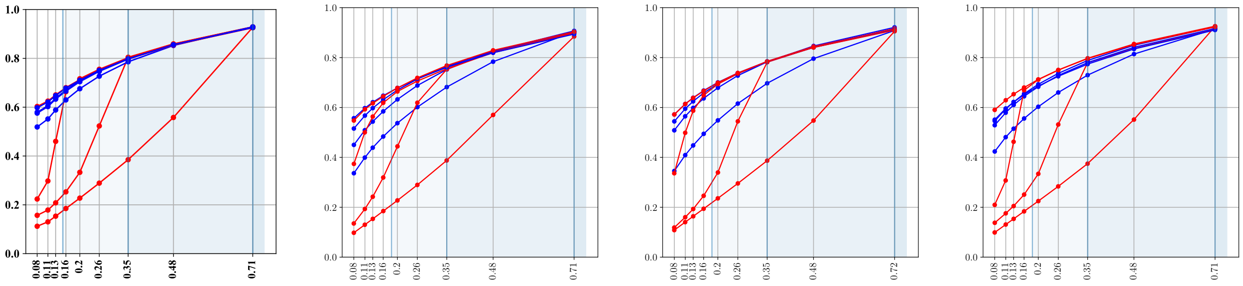


Figure A.1: First-order analysis of the residual of different architectures: Recurrent-CNN (Section A.1.2), UNet (Section A.1.3) and DenseNet (Section A.1.4) as a function of noise level. The plots show the magnitudes of the residual and the net bias averaged over 68 images in Set68 test set of Berkeley Segmentation Dataset (Martin et al., 2001) for networks trained over different training ranges. The range of noises used for training is highlighted in gray. (left) When the network is trained over the full range of noise levels ($\sigma \in [0, 100]$) the net bias is small, growing slightly as the noise increases. (middle and right) When the network is trained over the a smaller range ($\sigma \in [0, 55]$ and $\sigma \in [0, 30]$), the net bias grows explosively for noise levels outside the training range. This coincides with the dramatic drop in performance due to overfitting, reflected in the difference between the residual and the true noise.



(a) **(b)** **(c)** **(d)**

Figure A.2: Effect of net bias on performance for different architectures. Comparisons of architectures with (red curves) and without (blue curves) a net bias for the experimental design described in Section 2.3.3. The performance is quantified by the PSNR of the denoised image as a function of the input PSNR of the noisy image. All the architectures with bias perform poorly out of their training range, whereas the bias-free versions all achieve excellent generalization across noise levels. **(a)** Deep Convolutional Neural Network, DnCNN (Zhang et al., 2017a). **(b)** Recurrent architecture inspired by DURR (Zhang et al., 2018a). **(c)** Multiscale architecture inspired by the UNet (Ronneberger et al., 2015a). **(d)** Architecture with multiple skip connections inspired by the DenseNet (Huang et al., 2017).



(a) **(b)** **(c)** **(d)**

Figure A.3: Effect of net bias on performance in terms of SSIM for different architectures. Comparisons of architectures with (red curves) and without (blue curves) a net bias for the experimental design described in Section 2.3.3. The performance is quantified by the SSIM of the denoised image as a function of the input SSIM of the noisy image. All the architectures with bias perform poorly out of their training range, whereas the bias-free versions all achieve excellent generalization across noise levels. **(a)** Deep Convolutional Neural Network, DnCNN (Zhang et al., 2017a). **(b)** Recurrent architecture inspired by DURR (Zhang et al., 2018a). **(c)** Multiscale architecture inspired by the UNet (Ronneberger et al., 2015a). **(d)** Architecture with multiple skip connections inspired by the DenseNet (Huang et al., 2017).

A.4 Revealing the Denoising Mechanisms Learned by other architectures

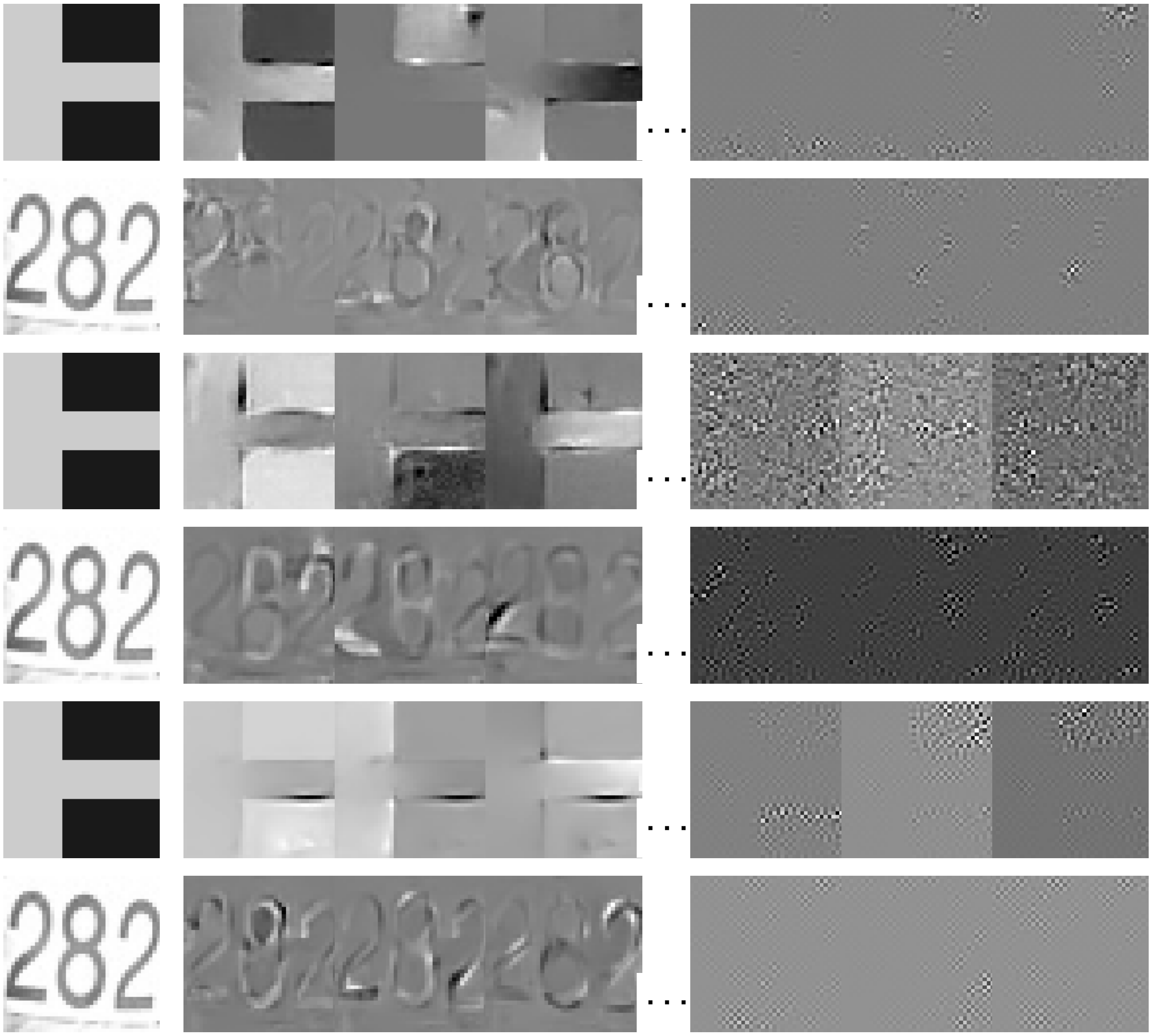


Figure A.4: Visualization of left singular vectors of the Jacobian for different architectures, a BF Recurrent CNN (top 2 rows), BF UNet (next 2 rows) and BF DenseNet (bottom 2 rows) evaluated on three different images, corrupted by noise with standard deviation $\sigma = 25$. The left column shows original (clean) images. The next three columns show singular vectors corresponding to non-negligible singular values. The vectors capture features from the clean image. The last three columns on the right show singular vectors corresponding to singular values that are almost equal to zero. These vectors are noisy and unstructured.

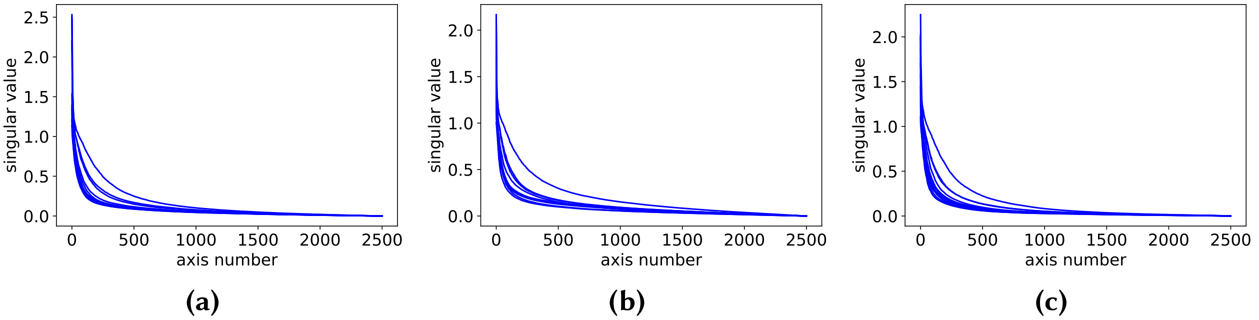


Figure A.5: Analysis of the SVD of the Jacobian of BF-CNN for ten natural images, corrupted by noise of standard deviation $\sigma = 50$. For all images, a large proportion of the singular values are near zero, indicating (approximately) a projection onto a subspace (the *signal subspace*). **(a)** Recurrent architecture inspired by DURR (Zhang et al., 2018a). **(b)** Multiscale architecture inspired by the UNet (Ronneberger et al., 2015a). **(c)** Architecture with multiple skip connections inspired by the DenseNet (Huang et al., 2017).

B | Generalization beyond memorizing training images

B.1 Experimental details

B.1.1 Training and architecture details

Architectures. We performed empirical experiments using two different architectures: UNet, and BF-CNN. All the denoisers are “bias-free”: we remove all additive constants from convolution and batch-normalization operations (i.e., the batch normalization does not subtract the mean). This facilitates unversality (denoisers can operate at all noise levels), and interpretability (network transformations are homogeneous of order 1, and the Jacobian provides a local characterization).

UNet networks contain 3 decoder blocks, one mid-level block, and 3 decoder blocks (Ronneberger et al., 2015b). Each block consists of 2 convolutional layers followed by a ReLU nonlinearity and bias-free batch-normalization. Each encoder block is followed by a 2 × 2 spacial down-sampling and a 2 fold increase in the number of channels. Each decoder block is followed by a 2 × 2 spacial upsampling and a 2 fold reduction of channels. The total number of parameters is $7.6m$.

BF-CNN networks Mohan* et al. (2020) are bias-free versions of DNCNN networks (Zhang et al., 2017b), 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. All convolutional kernels are of size 3 × 3, resulting in $700k$ parameters in total.

Training. We follow the training procedure described in Mohan* et al. (2020), minimizing the mean squared error in denoising 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, for 1000 epochs. 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 SISA sampling algorithm, which can operate without manual specification of the step size schedule (Kadkhodaie and Simoncelli, 2020). This method produces high-quality results in generative sampling, as well as sampling conditioned on linear measurements (Kadkhodaie and Simoncelli, 2021b).

Datasets. For experiments shown in Figures 3.2, 3.3, 3.5 and 3.13, we use the CelebA dataset (Liu et al., 2015) downsampled to 80 × 80 resolution. For experiments shown in Figure 3.7, we use images drawn from the LSUN bedroom dataset (Yu et al., 2015) downsampled to 80 × 80 resolution. This dataset is downsampled to 32 × 32 resolution for experiments shown in Figure 3.10. For experiments shown in Figure 3.8 we use CelebA HQ dataset (Karras et al., 2018) downsampled to 40 × 40 resolution.

B.2 Additional numerical results on inductive biases

B.2.1 More C^α examples

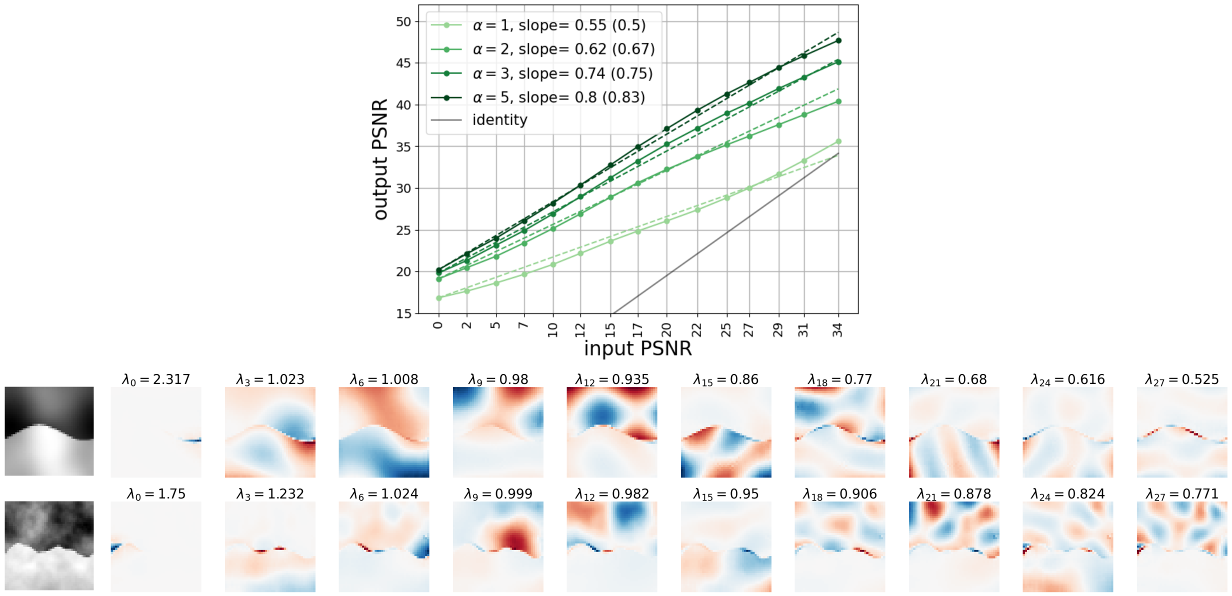


Figure B.1: BF-CNN denoisers trained on C^α images of size 40×40 achieve near-optimal performance. **Top.** PSNR curves of trained networks for various regularity levels α. The empirical slopes achieved for different values of α closely match the optimal slopes (dashed lines). **Bottom.** Eigenvectors for two C^α images (top row: $\alpha = 4$, bottom row: $\alpha = 2$), which consist of harmonics on the two regions and harmonics along the boundary. The frequency of the harmonics increases with k. For less regular images, the harmonics are more localized along the contours.

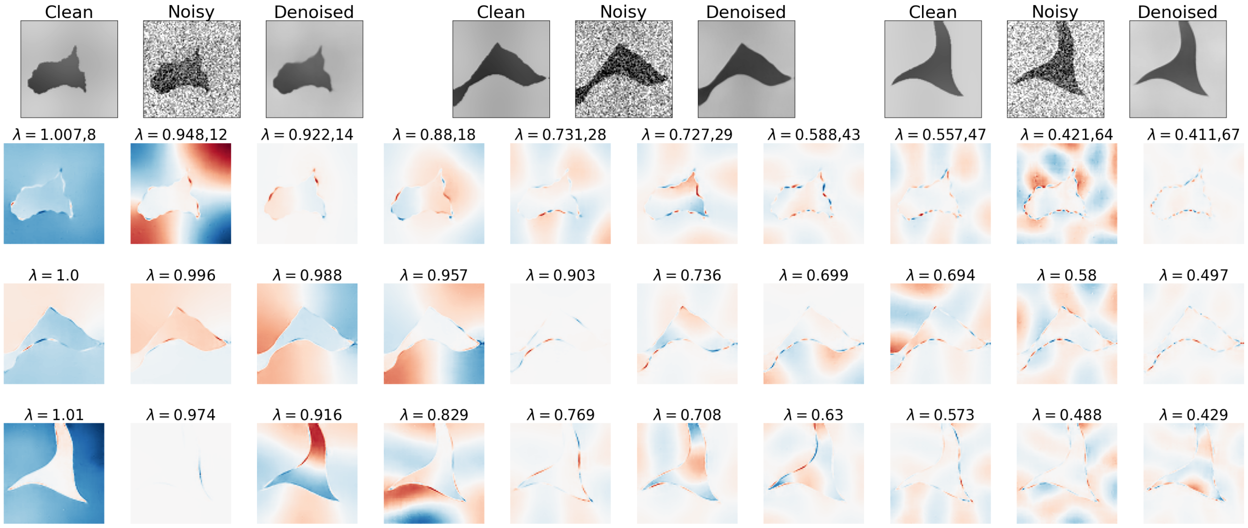


Figure B.2: Geometric-adaptive harmonic basis for three test images from C^α class. Here the regularity of the one-dimensional contours α_1 is different from the regularity of the two-dimensional background α_2. **Top.** Three example images. The regularity of the contour increases from left to right: $\alpha_1 = 1.5, 2, 4$. Background regularity is the same in all three examples, $\alpha_2 = 8$, and $\sigma = 0.2$. **Bottom.** Top 10 basis vectors for each image are shown. With increasing α_1, the contours become more regular, and the harmonics along the boundaries become less localized. This allows for a faster decay of coefficients and a lower denoising error.

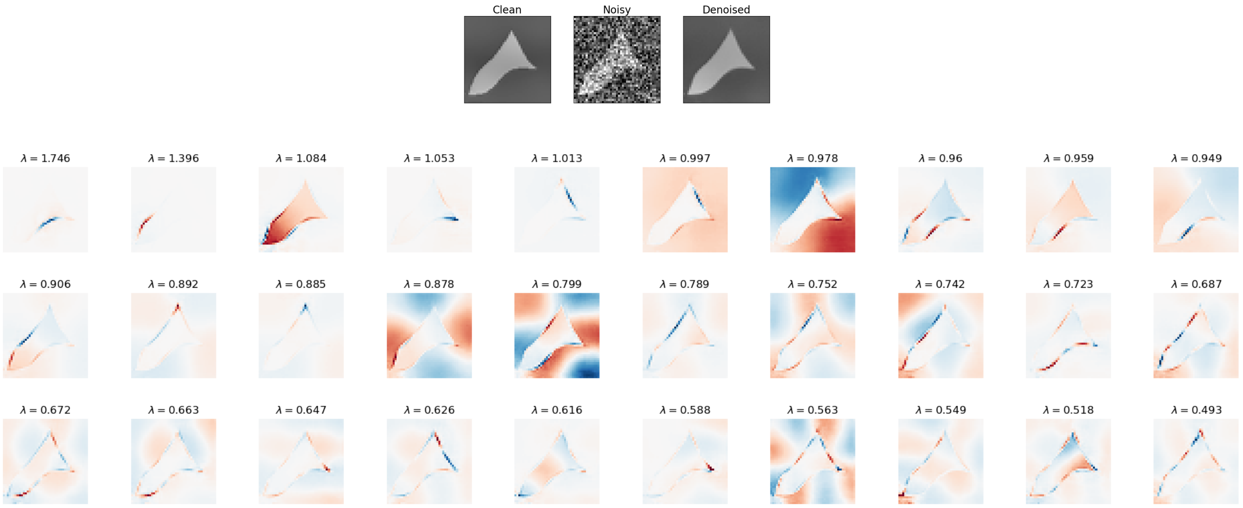


Figure B.3: Top. An additional example of a C^α test image with $\alpha = 3$. **Bottom.** Top eigenvectors of the geometric harmonic adaptive basis.

B.3 Mathematical derivations

B.3.1 Miyasawa relationships

The relationship of MMSE estimation of a signal corrupted by additive Gaussian noise to the score was published in (Miyasawa, 1961), and generalized in (Efron, 2011; Raphan and Simoncelli, 2011). For completeness, and notational consistency, we provide a derivation here. We begin by expressing the score $\nabla \log p(y)$ (dropping the σ dependence to simplify notation) and its Jacobian $\nabla^2 \log p(y)$ in terms of the measurement density $p(y|x)$ (which is Gaussian) and the posterior density $p(x|y)$. Using Bayes' rule and marginalization, the probability density of the noisy images is expressed as

$$p(y) = \int p(x)\, p(y|x)\, \mathrm{d}x.$$

Taking the logarithm and differentiating with respect to y, and using the fact that for any function h, $\nabla h(y) = h(y)\, \nabla \log h(y)$, we find

$$\begin{aligned} \nabla \log p(y) &= \int p(x)\, p(y|x)\, \nabla_y \log p(y|x)\, \mathrm{d}x \Big/ p(y) \\ &= \int p(x|y)\, \nabla_y \log p(y|x)\, \mathrm{d}x \\ &= \mathbb{E}\left[\nabla_y \log p(y|x) | y\right], \end{aligned} \tag{B.1}$$

which can be thought of as an equivalent of the chain rule on the scores as opposed to the densities.

Differentiating again with respect to y, we have

$$\nabla^2 \log p(y) = \int p(x|y)\, \left(\nabla_y \log p(x|y) \nabla_y \log p(y|x) \mathsf{T} + \nabla^2 \log p(y|x)\right) \mathrm{d}x. \tag{B.2}$$

The term $\nabla_y \log p(x|y)$ can be calculated by differentiating the logarithm of Bayes rule:

$$\begin{aligned} \log p(x|y) &= \log p(y|x) - \log p(y) + \log p(x), \\ \nabla_y \log p(x|y) &= \nabla_y \log p(y|x) - \nabla \log p(y), \end{aligned} \tag{B.3}$$

so that when injected into eq. (B.2) we obtain

$$\begin{aligned} \nabla^2 \log p(y) = \int & p(x|y) \left(\left(\nabla_y \log p(y|x) - \nabla \log p(y) \right) \nabla_y \log p(y|x)\mathsf{T} + \nabla^2 \log p(y) \right) \mathrm{d}x \\ &= \mathbb{E} \left[\left(\nabla_y \log p(y|x) - \nabla \log p(y) \right) \nabla_y \log p(y|x)\mathsf{T} | y \right] + \mathbb{E} \left[\nabla^2 \log p(y|x) | y \right] \\ &= \mathrm{Cov}[\nabla_y \log p(y|x) | y] + \mathbb{E} \left[\nabla^2 \log p(y|x) | y \right], \end{aligned} \tag{B.4}$$

where the last line used $\nabla \log p(y) = \mathbb{E} \left[\nabla_y \log p(y|x) | y \right]$.

We then use the fact that y is obtained from x by adding Gaussian white noise of variance $\sigma^2 \mathrm{Id}$:

$$\log p(y|x) = -\frac{1}{2\sigma^2} \|y - x\|^2 + \mathrm{cst}, \tag{B.5}$$

$$\nabla_y \log p(y|x) = -\frac{1}{\sigma^2} (y - x), \tag{B.6}$$

$$\nabla_y^2 \log p(y|x) = -\frac{1}{\sigma^2} \mathrm{Id}, \tag{B.7}$$

so that eqs. (B.1) and (B.4) become

$$\begin{aligned} \nabla \log p(y) &= \frac{1}{\sigma^2} \left(\mathbb{E}[x|y] - y \right), \\ \nabla^2 \log p(y) &= \frac{1}{\sigma^4} \mathrm{Cov}[x|y] - \frac{1}{\sigma^2} \mathrm{Id}. \end{aligned}$$

Finally, the above identities can be rearranged to yield the first- and second-order Miyasawa relationships:

$$\mathbb{E}\left[x|y\right] = y + \sigma^2 \nabla \log p(y), \tag{B.8}$$

$$\mathrm{Cov}[x|y] = \sigma^2 \left(\mathrm{Id} + \sigma^2 \nabla^2 \log p(y)\right). \tag{B.9}$$

Note that the optimal denoising error satisfies

$$\mathbb{E}\left[\|x - \mathbb{E}\left[x|y\right]\|^2\right] = \mathbb{E}\left[\mathbb{E}\left[\mathrm{tr}(x - \mathbb{E}\left[x|y\right])(x - \mathbb{E}\left[x|y\right])^T|y\right]\right] = \mathbb{E}\left[\mathrm{tr}\,\mathrm{Cov}\left[x|y\right]\right].$$

B.3.2 Control on Kullback-Leibler divergence

Equation (3.1) results from Theorem 1 of Song et al. (2021a), considering the so-called "variance-exploding" SDE $\mathrm{d}x_t = \mathrm{d}w_t$ where $(w_t)_{t\geq 0}$ is a Brownian motion ($t = \sigma^2$ then corresponds to the noise variance), and letting the stopping time T go to infinity.

To reformulate the score-matching error as a denoising objective, we insert the Miyasawa equation (2.3) as well as the expression of the score model $s_\theta(y) = (f_\theta(y) - y)/\sigma^2$ into the score-matching error:

$$\mathbb{E}\left[\|\nabla \log p_\sigma(y) - s_\theta(y)\|^2\right] = \frac{1}{\sigma^4}\mathbb{E}\left[\|\mathbb{E}[x|y] - f_\theta(y)\|^2\right] \tag{B.10}$$

We recall the decomposition of the denoising error when conditioning on y:

$$\mathbb{E}\left[\|x - f_\theta(y)\|^2\right] = \mathbb{E}\left[\|x - \mathbb{E}[x|y]\|^2\right] + \mathbb{E}\left[\|\mathbb{E}[x|y] - f_\theta(y)\|^2\right], \tag{B.11}$$

so that inserting eq. (B.11) into eq. (B.10) yields

$$\begin{aligned}\mathbb{E}\left[\|\nabla \log p_\sigma(y) - s_\theta(y)\|^2\right] &= \frac{1}{\sigma^4}\left(\mathbb{E}\left[\|x - f_\theta(y)\|^2\right] - \mathbb{E}\left[\|x - \mathbb{E}[x|y]\|^2\right]\right) \\ &= \frac{1}{\sigma^4}\left(\mathrm{MSE}(f_\theta, \sigma^2) - \mathrm{MSE}(f^\star, \sigma^2)\right).\end{aligned}$$

Combined with eq. (3.1), this proves eq. (3.3).

B.3.3 SURE objective

We decompose the MSE as follows:

$$\begin{aligned}\mathbb{E}\left[\|x - f(y)\|^2\right] &= \mathbb{E}\left[\|(y - f(y)) - (y - x)\|^2\right] \\ &= \mathbb{E}\left[\|y - f(y)\|^2\right] - 2\mathbb{E}[\langle y - x, y - f(y)\rangle] + \mathbb{E}\left[\|y - x\|^2\right]. \end{aligned} \tag{B.12}$$

The last term is the total variance of the noise and is thus equal to $\sigma^2 d$. The middle term can be rewritten with an integration by parts, using the fact that $y - x = -\sigma^2 \nabla_y \log p(y|x)$:

$$\begin{aligned}\mathbb{E}[\langle y - x, y - f(y)\rangle] &= -\sigma^2 \iint \left\langle \nabla_y \log p(y|x), y - f(y) \right\rangle \, p(x)\, p(y|x) \, \mathrm{d}x \, \mathrm{d}y, \\ &= -\sigma^2 \iint \left\langle \nabla_y p(y|x), y - f(y) \right\rangle \, p(x) \, \mathrm{d}x \, \mathrm{d}y, \\ &= \sigma^2 \iint \operatorname{tr}\left(\mathrm{Id} - \nabla f(y)\right) \, p(x)\, p(y|x) \, \mathrm{d}x \, \mathrm{d}y, \\ &= \sigma^2 \mathbb{E}[d - \operatorname{tr} \nabla f(y)]. \end{aligned} \tag{B.13}$$

Inserting eq. (B.13) into eq. (B.12), we then obtain

$$\mathbb{E}\left[\|x - f(y)\|^2\right] = \mathbb{E}\left[\|y - f(y)\|^2\right] + 2\sigma^2 \mathbb{E}[\operatorname{tr} \nabla f(y)] - \sigma^2 d, \tag{B.14}$$

proving the Stein's Unbiased Risk Estimator of the MSE.

B.3.4 Optimal thresholding in a basis

For completeness, we derive here the error of the fixed-basis oracle denoiser (Donoho, 1995; Donoho and Johnstone, 1994; Mallat, 2008).

We consider an oracle denoiser which computes

$$\sum_k \lambda_k(x)\ \langle y, e_k\rangle\ e_k.$$

In practice, the denoiser does not have access to the clean image x, and the shrinkage factors λ_k thus have to be estimated from the noisy image y alone. Note however that optimizing this oracle estimator is non-trivial as the shrinkage factors have to be independent from the noise.

We can then compute its denoising error on a clean image x by averaging over the noise

$$\begin{aligned}\mathbb{E}\left[\left\|x - \sum_k \lambda_k(x)\ \langle y, e_k\rangle\ e_k\right\|^2 |x\right] &= \mathbb{E}\left[\sum_k \left(\langle x, e_k\rangle - \lambda_k(x)\ \langle y, e_k\rangle\right)^2 |x\right] \\ &= \mathbb{E}\left[\sum_k \left((1-\lambda_k(x))\,\langle x, e_k\rangle - \lambda_k(x)\,\langle z, e_k\rangle\right)^2 |x\right] \\ &= \sum_k \left((1-\lambda_k(x))^2\,\langle x, e_k\rangle^2 + \lambda_k(x)^2\sigma^2\right), \qquad \text{(B.15)}\end{aligned}$$

where the last step used the fact that $\langle z, e_k\rangle \sim \mathcal{N}(0, \sigma^2)$ independently from x. For each x and k, the optimal oracle shrinkage factor $\lambda_k(x)$ thus minimizes the quadratic function

$$(1-\lambda_k(x))^2\,\langle x, e_k\rangle^2 + \lambda_k(x)^2\sigma^2,$$

which is achieved when

$$\lambda_k(x) = \frac{\langle x, e_k\rangle^2}{\langle x, e_k\rangle^2 + \sigma^2}. \qquad \text{(B.16)}$$

Injecting eq. (B.16) into eq. (B.15) gives the denoising error on x as

$$\mathbb{E}\left[\left\|x - \sum_k \lambda_k(x)\ \langle y, e_k\rangle\ e_k\right\|^2 |x\right] = \sum_k \frac{\sigma^2\,\langle x, e_k\rangle^2}{\langle x, e_k\rangle^2 + \sigma^2}. \qquad \text{(B.17)}$$

Incidentally, this error is also equal to $\sigma^2 \sum_k \lambda_k(x)$, similarly to the optimal denoiser as shown in eq. (3.8).

The fraction $\frac{\sigma^2 \langle x, e_k \rangle^2}{\langle x, e_k \rangle^2 + \sigma^2}$ is of the same order as $\min(\langle x, e_k \rangle^2, \sigma^2)$ up to a factor of 2, as we have the inequalities for any $a, b > 0$

$$\frac{1}{2} \min(a, b) \leq \frac{ab}{a+b} \leq \min(a, b),$$

which follow from $ab = \min(a, b) \max(a, b)$ and $\max(a, b) \leq a + b \leq 2 \max(a, b)$. We thus have

$$\begin{aligned} \mathbb{E}\left[\left\| x - \sum_k \lambda_k(x) \, \langle y, e_k \rangle \, e_k \right\|^2 \middle| x \right] &\sim \sum_k \min\left(\langle x, e_k \rangle^2, \sigma^2\right) \\ &= \sum_{\langle x, e_k \rangle^2 > \sigma^2} \sigma^2 + \sum_{\langle x, e_k \rangle^2 < \sigma^2} \langle x, e_k \rangle^2 . \end{aligned} \tag{B.18}$$

Let M be the number of terms in the left sum (that is, the number of ranks k such that $\langle x, e_k \rangle^2 > \sigma^2$), and $x_M = \sum_{\langle x, e_k \rangle^2 > \sigma^2} \langle x, e_k \rangle \, e_k$ be the M-term approximation of x. We then have

$$\| x - x_M \|^2 = \left\| \sum_{\langle x, e_k \rangle^2 < \sigma^2} \langle x, e_k \rangle \, e_k \right\| = \sum_{\langle x, e_k \rangle^2 < \sigma^2} \langle x, e_k \rangle^2 , \tag{B.19}$$

so that plugging eq. (B.19) into eq. (B.18) gives

$$\mathbb{E}\left[\left\| x - \sum_k \lambda_k(x) \, \langle y, e_k \rangle \, e_k \right\|^2 \middle| x \right] \sim M\sigma^2 + \| x - x_M \|^2. \tag{B.20}$$

This realizes a decomposition of the oracle denoising error into a denoising bias $\| x - x_M \|^2$, which corresponds to the signal variance that has been lost, and a denoising variance $M\sigma^2$, which corresponds to the noise variance that has been preserved (note that denoising bias and variance are different than the model variance and model bias studied in the paper). The sum of the two terms captures the efficiency of the approximation of x in the basis (e_k).

Let us reorder the coefficients so that $\langle x, e_1 \rangle^2 \geq \cdots \geq \langle x, e_d \rangle^2$(note that the ordering depends on x), and assume that $\langle x, e_k \rangle^2 \sim k^{-(\alpha+1)}$ for some $\alpha > 0$. More precisely, we assume that there exists two constants c, c' independent of x and k such that $c\, k^{-(\alpha+1)} \leq \langle x, e_k \rangle^2 \leq c'\, k^{-(\alpha+1)}$. By definition of M,

$$\langle x, e_M \rangle^2 > \sigma^2 \geq \langle x, e_{M+1} \rangle^2,$$

so that

$$c'\, M^{-(\alpha+1)} > \sigma^2 \geq c\, (M+1)^{-(\alpha+1)}.$$

We then have $M^{-(\alpha+1)} \sim \sigma^2$, i.e., $M \sim \sigma^{-2/(\alpha+1)}$, and thus $M\sigma^2 \sim \sigma^{2\alpha/(\alpha+1)}$. We also have

$$\begin{aligned}
\sum_{k>M} \langle x, e_k \rangle^2 &\leq c' \sum_{k>M} k^{-(\alpha+1)} \leq c' \int_M^{+\infty} t^{-(\alpha+1)} \mathrm{d}t = \frac{c'}{\alpha} M^{-\alpha}, \\
\sum_{k>M} \langle x, e_k \rangle^2 &\geq c \sum_{k>M} k^{-(\alpha+1)} \geq c \int_{M+1}^{+\infty} t^{-(\alpha+1)} \mathrm{d}t = \frac{c}{\alpha} (M+1)^{-\alpha},
\end{aligned}$$

so that $\|x - x_M\|^2 \sim M^{-\alpha} \sim \sigma^{2\alpha/(\alpha+1)}$. Finally, we have shown that the two terms in eq. (B.20) are of the same order, and it follows that

$$\mathbb{E}\left[\left\| x - \sum_k \lambda_k(x)\ \langle y, e_k \rangle\ e_k \right\|^2 \middle| x \right] \sim M\sigma^2 + \|x - x_M\|^2 \sim \sigma^{2\alpha/(\alpha+1)}.$$

Because the constants have been assumed to be independent of x, one can average over x to obtain that the oracle MSE is $\sim \sigma^{2\alpha/(\alpha+1)}$.

B.4 Geometric C$^\alpha$ images

A continuous image $x \colon [0,1]^2 \to \mathbb{R}$ is part of the geometric C$^\alpha$ class (Donoho, 1999; Korostelev and Tsybakov, 1993; Peyré and Mallat, 2008) if it is uniformly α-Lipschitz over $[0,1]^2 \setminus \{\gamma_i\}$, where the γ_i are uniformly α-Lipschitz curves in $[0,1]^2$ which do not intersect tangentially. A function f is uniformly α-Lipschitz over a domain Ω if there exists a constant C such that for all $x \in \Omega$, there exists a polynomial q_x of degree $\lfloor \alpha \rfloor$ such that for all $y \in \Omega$,

$$|f(y) - q_x(y)| \leq C\,|x-y|^\alpha. \tag{B.21}$$

We explain how to generate numerically such images in Algorithm 4.

Algorithm 4: Synthesis of a $\mathbf{C}^\alpha$ image via integration

```
requires: regularity $\alpha$, Fast Fourier Transform FFT
Make a contour
Define a 1D filter $f_1(\omega) = |\omega|^{-\alpha}$
Draw a random 1D $\mathbf{C}^0$ function with i.i.d.  uniform entries $c(t) \sim \mathcal{U}([-0.5, 0.5])$
Integrate in the Fourier domain to define $C = \mathrm{FFT}^{-1}(f_1 \times \mathrm{FFT}(c))$
Make the background
Define a 2D filter $f_2(\omega) = (\omega_1^2 + \omega_2^2)^{-\alpha/2}$
Draw two random 2D $\mathbf{C}^0$ functions with i.i.d. uniform entries
  $b_1(x,y), b_2(x,y) \sim \mathcal{U}([-0.5, 0.5])$
Integrate in the Fourier domain to define $B_i = \mathrm{FFT}^{-1}(f_2 \times \mathrm{FFT}(b_i))$ $(i = 1, 2)$
Make a mask and combine
Define a binary mask $M = \mathbb{1}_{y>C}$
Let $x = M \times B_1 + (1 - M) \times B_2$
return $x$
```

C | Muli-scale Local conditional image density

C.1 Proof of Theorem 4.7

To simplify notation, we drop the j subscript. Let I (resp. J) denote the set of indices of pixel values of $\bar{\boldsymbol{x}}$ (resp. $\boldsymbol{x}$). If S is a set of indices, we denote $\bar{\boldsymbol{x}}(S) = (\bar{\boldsymbol{x}}(i))_{i \in S \cap I}$. 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 4.7, we need to show that the local Markov property:

$$\forall i \in I,\ p\big(\bar{\boldsymbol{x}}(i) \,\big|\, \bar{\boldsymbol{x}}(I \setminus \{i\}), \boldsymbol{x}\big) = p\big(\bar{\boldsymbol{x}}(i) \,\big|\, \bar{\boldsymbol{x}}(N(i)), \boldsymbol{x}(N_+(i))\big), \tag{C.1}$$

is equivalent to the conditional score being computable with RFs restricted to neighborhoods:

$$\forall i \in I,\ \frac{\partial \log p}{\partial \bar{\boldsymbol{x}}(i)}\big(\bar{\boldsymbol{x}} \,\big|\, \boldsymbol{x}\big) = f_i\big(\bar{\boldsymbol{x}}(N_+(i)), \boldsymbol{x}(N_+(i))\big), \tag{C.2}$$

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:

$$\begin{aligned} p\big(\bar{x} \,\big|\, \boldsymbol{x}\big) &= p\big(\bar{x}(i) \,\big|\, \bar{x}(I \setminus \{i\}), \boldsymbol{x}\big)\, p\big(\bar{x}(I \setminus \{i\}) \,\big|\, \boldsymbol{x}\big) \\ &= p\big(\bar{x}(i) \,\big|\, \bar{x}(N(i)), \boldsymbol{x}(N_+(i))\big)\, p\big(\bar{x}(I \setminus \{i\}) \,\big|\, \boldsymbol{x}\big), \end{aligned}$$

where we have used eq. (C.1) in the last step. Then, taking the logarithm and differentiating, only the first term remains:

$$\frac{\partial \log p}{\partial \bar{x}(i)}\big(\bar{x} \,\big|\, \boldsymbol{x}\big) = \frac{\partial \log p}{\partial \bar{x}(i)}\big(\bar{x}(i) \,\big|\, \bar{x}(N(i)), \boldsymbol{x}(N_+(i))\big),$$

which proves eq. (C.2).

Reciprocally, we now prove that eq. (C.2) implies eq. (C.1). Let $i \in I$, and $\boldsymbol{\delta}_i(j) = \delta_{ij}$ where δ_{ij} is the Kronecker delta. We have, by integrating the partial derivative:

$$\begin{aligned} \log p\big(\bar{x} \,\big|\, \boldsymbol{x}\big) &= \log p\big(\bar{x} - \bar{x}(i)\boldsymbol{\delta}_i \,\big|\, \boldsymbol{x}\big) - \int_0^1 \frac{\partial \log p}{\partial \bar{x}(i)}\big(\bar{x} - t\bar{x}(i)\boldsymbol{\delta}_i \,\big|\, \boldsymbol{x}\big)\mathrm{d}t \\ &= \log p\big(\bar{x} - \bar{x}(i)\boldsymbol{\delta}_i \,\big|\, \boldsymbol{x}\big) - \int_0^1 f_i\big(\bar{x}(N_+(i)) - t\bar{x}(i)\boldsymbol{\delta}_i, \boldsymbol{x}(N_+(i))\big)\mathrm{d}t, \end{aligned}$$

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 $\boldsymbol{x}(N_+(i))$. This implies that when we condition on $\bar{x}(N(i))$ and $\boldsymbol{x}(N_+(i))$, 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 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., 2020). 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-multi-scale) CNNs for denoising. For RF 43×43, we used the same architecture as for the coarsest scale band of the multi-scale 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 Figure 4.10 and Figure 4.11, 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 Figure 4.12, Figure C.1 and Figure 4.13, 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., 2020), 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, 2021b).

C.3 Additional Super-resolution Examples

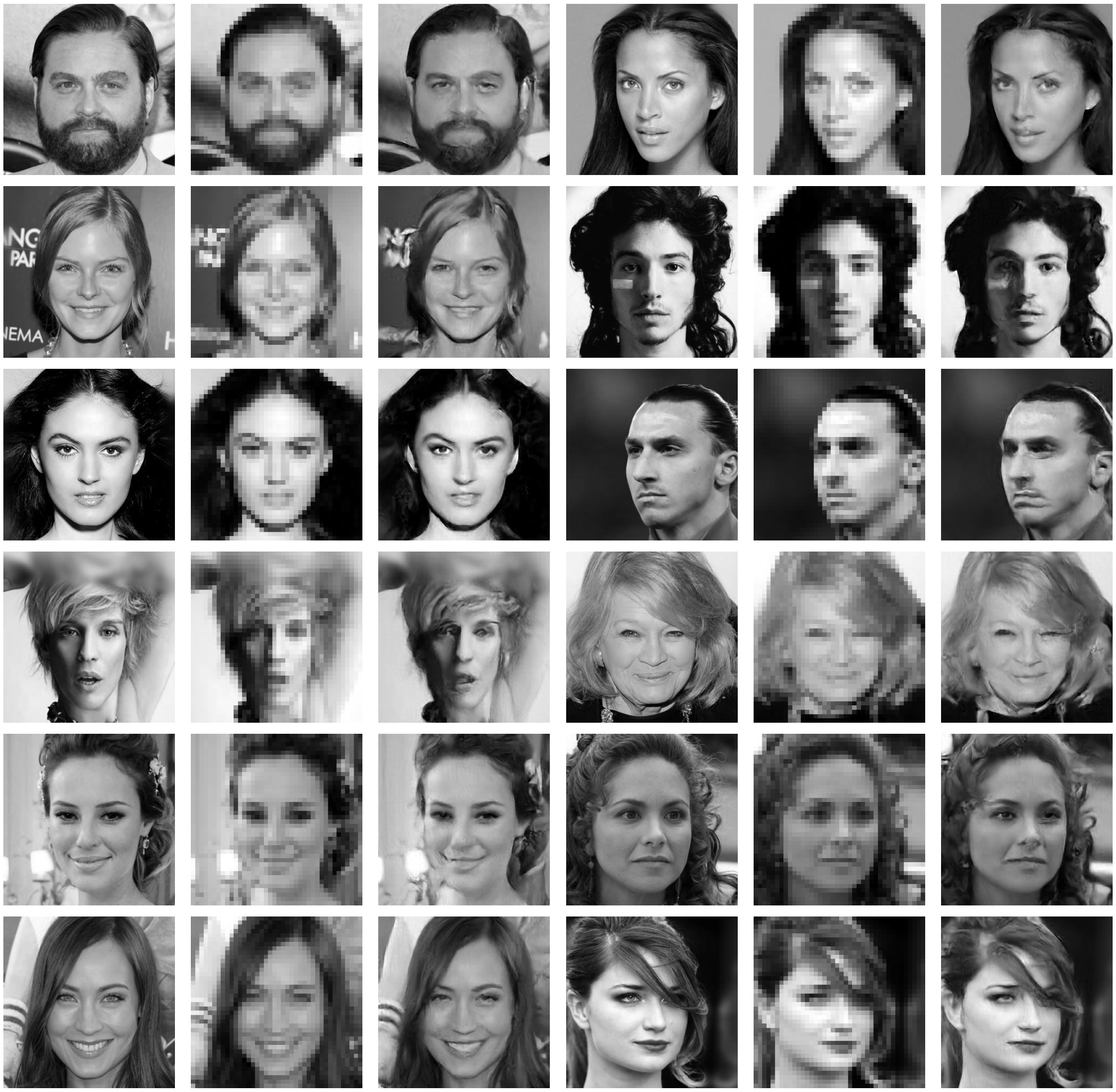


Figure C.1: Additional super-resolution examples. **Columns, in groups of three, from left to right:** original images, 320 × 320 pixels. Low-pass images of corresponding 3-stage wavelet decomposition (downsampled to 40 × 40, expanded to full size for viewing). Coarse-to-fine conditional generation using the multi-scale conditional denoisers, each with RF size 13 × 13.

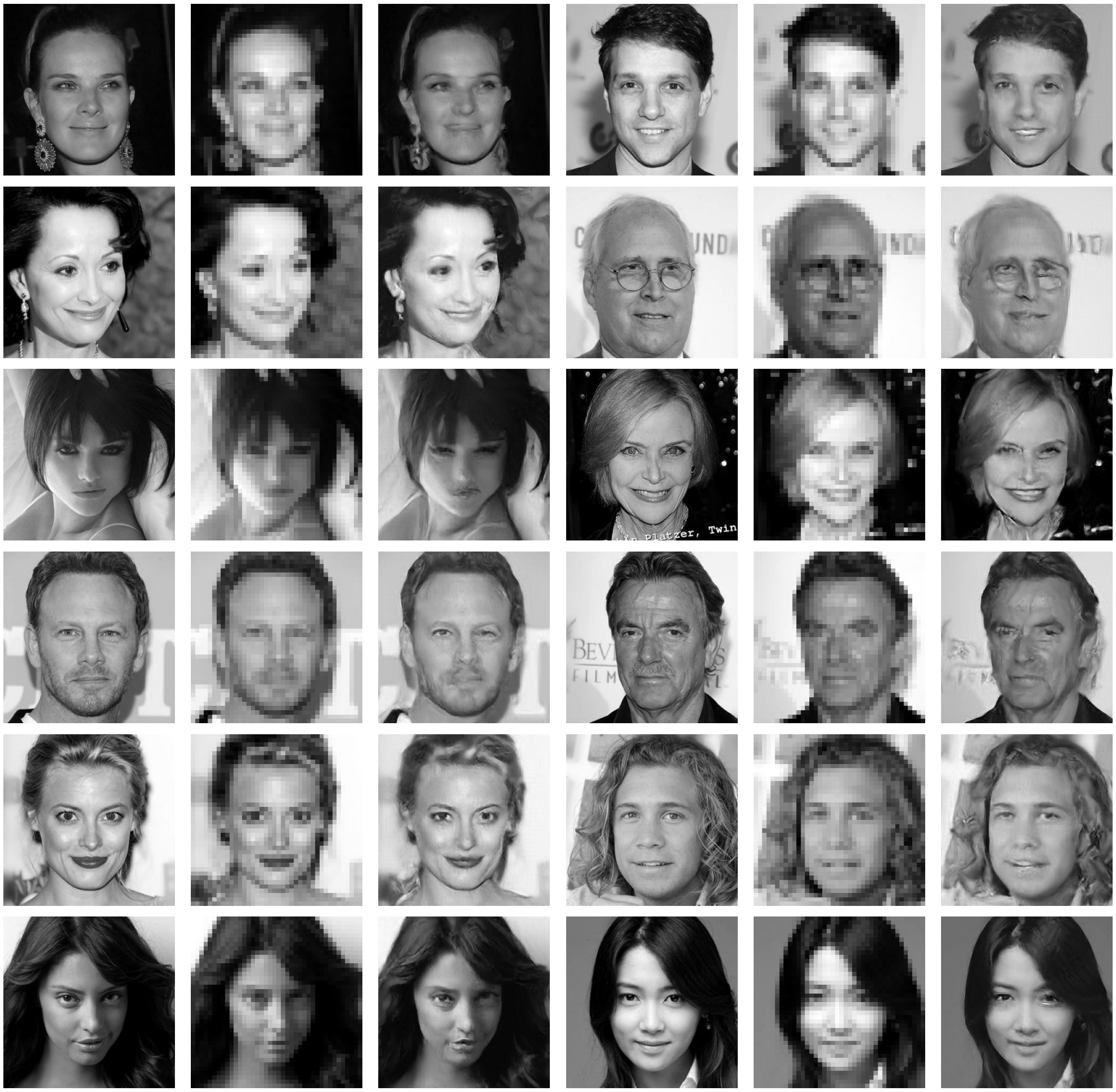


Figure C.2: Additional super-resolution examples. See caption of Fig. C.1.

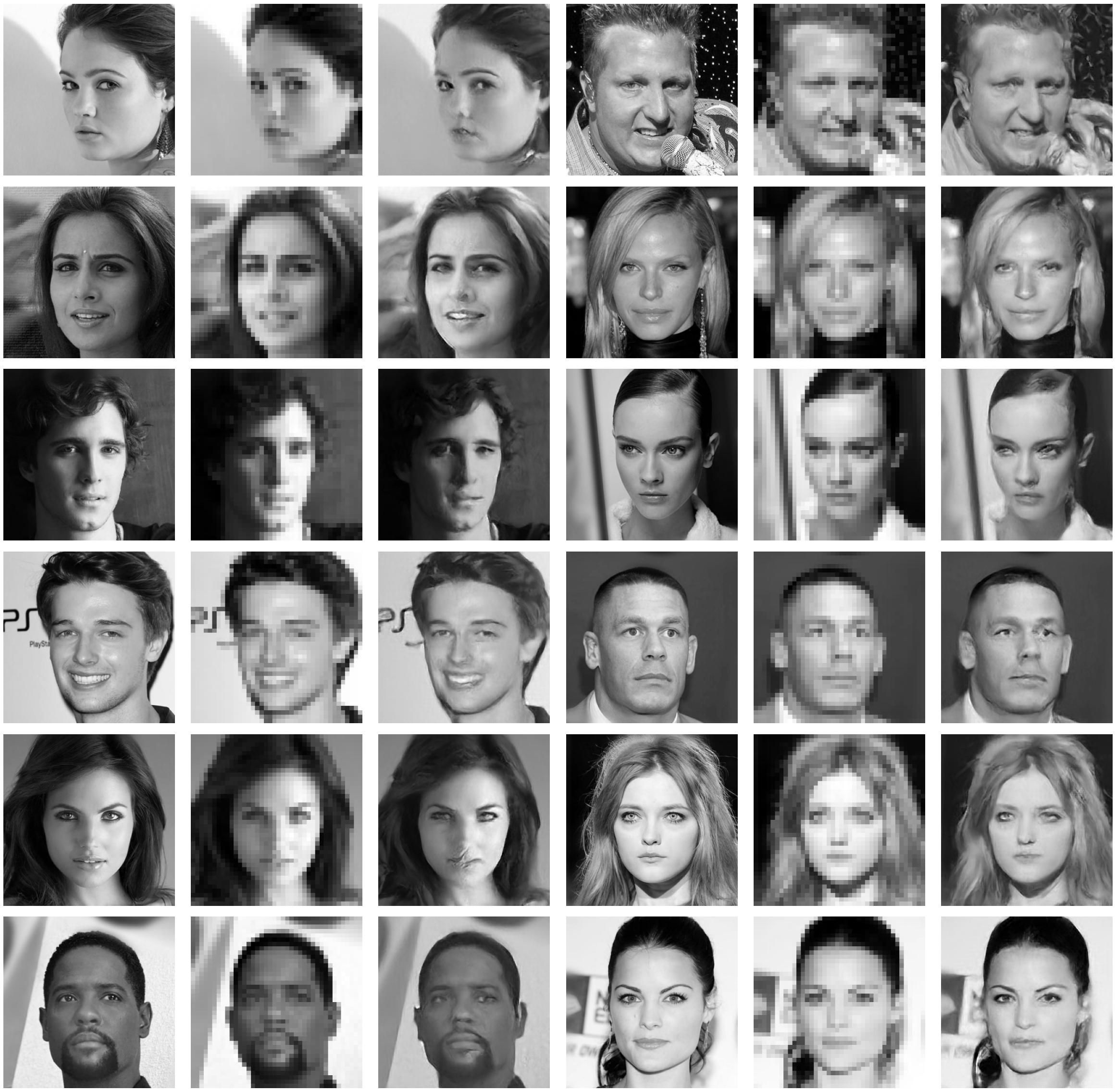


Figure C.3: Additional super-resolution examples. See caption of Fig. C.1.

Bibliography

Alain, G. and Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593.

Ballé, J., Laparra, V., and Simoncelli, E. P. (2016). Density modeling of images using a generalized normalization transformation. In *Int'l Conf on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Barron, A., Birgé, L., and Massart, P. (1999). Risk bounds for model selection via penalization. *Probability theory and related fields*, 113:301–413.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013). Generalized denoising auto-encoders as generative models. In *Adv. Neural Information Processing Systems (NIPS*13)*, pages 899–907. MIT Press.

Bereska, L. and Gavves, E. (2024). Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):192–225.

Biroli, G., Bonnaire, T., de Bortoli, V., and Mézard, M. (2024). Dynamical regimes of diffusion models. *arXiv preprint arXiv:2402.18491*.

Blake, A. (1989). Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 11(1):2–12.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction*. MIT Press.

Blumensath, T. and Davies, M. E. (2009). Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 55(4):1872–1882.

Bouman, C. and Liu, B. (1988). Segmentation of textured images using a multiple resolution approach. In *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, pages 1124–1127 vol.2.

Bouman, C. A. and Sauer, K. (1994). Maximum likelihood scale estimation for a class of markov random fields. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages V–537. IEEE.

Bouman, C. A. and Shapiro, M. (1994). A multiscale random field model for bayesian image segmentation. *IEEE Transactions on image processing*, 3(2):162–177.

Buccigrossi, R. W. and Simoncelli, E. P. (1997). Progressive wavelet image coding based on a conditional probability model. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2957–2960. IEEE.

Buccigrossi, R. W. and Simoncelli, E. P. (1999a). Image compression via joint statistical characterization in the wavelet domain. *IEEE transactions on Image processing*, 8(12):1688–1701.

Buccigrossi, R. W. and Simoncelli, E. P. (1999b). Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans Image Processing*, 8(12):1688–1701.

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). Image denoising: Can plain neural networks compete with bm3d? In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2392–2399. IEEE.

Burt, P. J. and Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. COM-31(4):532–540.

Candès, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. (2023). Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270.

Chambolle, A., DeVore, R. A., Lee, N., and Lucier, B. J. (1998). Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans Image Processing*, 7:319–335.

Chan, S. H., Wang, X., and Elgendy, O. A. (2017). Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98.

Chen, Y. and Pock, T. (2017). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Patt. Analysis and Machine Intelligence*, 39(6):1256–1272.

Choi, S., Isidoro, J., Getreuer, P., and Milanfar, P. (2018). Fast, trainable, multiscale denoising. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 963–967. IEEE.

Clifford, P. and Hammersley, J. (1971). Markov fields on finite graphs and lattices.

Cottaris, N. P., Jiang, H., Ding, X., Wandell, B. A., and Brainard, D. H. (2019). A computational-observer model of spatial contrast sensitivity: Effects of wave-front-based optics, cone-mosaic structure, and inference engine. *Journal of vision*, 19(4):8–8.

Crouse, M. S., Nowak, R. D., and Baraniuk, R. (1998). Wavelet-based statistical signal processing using hidden markov models. *IEEE Trans Signal Processing*, 46(4):886–902.

Şendur, L. and Selesnick, I. W. (2002). Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans Signal Proc*, 50(11):2744–2756.

Cui, Y. Q. and Wang, K. (2005). 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.

Dar, S. U. H., Ghanaat, A., Kahmann, J., Ayx, I., Papavassiliou, T., Schoenberg, S. O., and Engelhardt, S. (2023). Investigating data memorization in 3d latent diffusion models for medical image synthesis. *arXiv preprint arXiv:2307.01148*.

Dobrushin, R. L. (1968). The description of the random field by its conditional distributions and its regularity conditions. *Teoriya Veroyatnostei i ee Primeneniya*, 13(2):201–229.

Donoho, D. (1995). Denoising by soft-thresholding. *IEEE Trans Info Theory*, 43:613–627.

Donoho, D. L. (1999). Wedgelets: Nearly minimax estimation of edges. *the Annals of Statistics*, 27(3):859–897.

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans Info Theory*, 52(4):1289–1306.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455.

Dossal, C., Le Pennec, E., and Mallat, S. (2011). bandlet image estimation with model selection. *Signal Processing*, 91:2743–2753.

Efron, B. (2011). Tweedie's formula and selection bias. *J American Statistical Association*, 106(496):1602–1614.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image processing*, 15(12):3736–3745.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. (2021). A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pages 877–894.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Josa a*, 4(12):2379–2394.

Freeman, W. and Liu, C. (2011). Markov random fields for super-resolution and texture synthesis. *Advances in Markov Random Fields for Vision and Image Processing*, 1(155-165):3.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., and Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Geman, D. and Reynolds, G. (1992). Constrained restoration and the recovery of discontinuities. *IEEE Transactions on pattern analysis and machine intelligence*, 14(3):367–383.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Geman, S., McClure, D. E., and Geman, D. (1992). A nonlinear filter for film restoration and other problems in image processing. *CVGIP: Graphical models and image processing*, 54(4):281–289.

Goyal, A. and Bengio, Y. (2022). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068.

Griffiths, T. L., Zhu, J.-Q., Grant, E., and McCoy, R. T. (2023). Bayes in the age of intelligent machines. *arXiv preprint arXiv:2311.10206*.

Guth, F., Coste, S., De Bortoli, V., and Mallat, S. (2022a). Wavelet score-based generative modeling. *arXiv preprint arXiv:2208.05003*.

Guth, F., Coste, S., De Bortoli, V., and Mallat, S. (2022b). Wavelet score-based generative modeling. *arXiv preprint arXiv:2208.05003*.

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Math Annal.*, 69:331–371.

Hamm, J. and Lee, D. D. (2008). Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th international conference on Machine learning*, pages 376–383.

Hebert, T. and Leahy, R. (1989). A generalized em algorithm for 3-d bayesian reconstruction from poisson data using gibbs priors. *IEEE transactions on medical imaging*, 8(2):194–202.

Hel-Or, Y. and Shaked, D. (2008). A discriminative approach for wavelet shrinkage denoising. *IEEE Trans. Image Processing*, 17(4).

Herbreteau, S., Moebel, E., and Kervrann, C. (2024). Normalization-equivariant neural networks with application to image denoising. *Advances in Neural Information Processing Systems*, 36.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. (2022). Cascaded diffusion models for high fidelity image generation. *J Machine Learning Research*, 23:47–1.

Hofer, H., Carroll, J., Neitz, J., Neitz, M., and Williams, D. R. (2005). Organization of the human trichromatic cone mosaic. *Journal of Neuroscience*, 25(42):9669–9679.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4700–4708.

Hyvärinen, A. and Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Ilesanmi, A. E. and Ilesanmi, T. O. (2021). Methods for image denoising using convolutional neural network: a review. *Complex & Intelligent Systems*, 7(5):2179–2198.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jain, V. and Seung, S. (2009). Natural image denoising with convolutional networks. In *Advances in neural information processing systems*, pages 769–776.

Kadkhodaie, Z., Guth, F., Mallat, S., and Simoncelli, E. P. (2023a). Learning multi-scale local conditional probability models of images. In *Int'l Conf on Learning Representations (ICLR)*, Kigali, Rwanda.

Kadkhodaie, Z., Guth, F., Simoncelli, E. P., and Mallat, S. (2023b). Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*.

Kadkhodaie, Z. and Simoncelli, E. (2021a). Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254.

Kadkhodaie, Z. and Simoncelli, E. P. (2020). Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*.

Kadkhodaie, Z. and Simoncelli, E. P. (2021b). Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Adv Neural Information Processing Systems (NeurIPS*21)*, volume 34.

Kamilov, U. S., Mansour, H., and Wohlberg, B. (2017). A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters*, 24(12):1872–1876.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Kato, Z. (2008). Segmentation of color images via reversible jump mcmc sampling. *Image and Vision Computing*, 26(3):361–371.

Kato, Z., Zerubia, J., et al. (2012). Markov random fields in image segmentation. *Foundations and Trends® in Signal Processing*, 5(1–2):1–155.

Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and Their Applications*. Contemporary Mathematics, DOI: http://dx.doi.org/10.1090/conm/001.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*, 50:363–370.

Korostelev, A. P. and Tsybakov, A. B. (1993). *Minimax theory of image reconstruction*. Springer New York, NY.

Laumont, R., Bortoli, V. D., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2021). Bayesian imaging using plug & play priors: When Langevin meets Tweedie. *arXiv preprint arXiv:2103.04715v4.*

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

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lemke, C., Budka, M., and Gabrys, B. (2015). Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44:117–130.

Li, C., Yin, W., Jiang, H., and Zhang, Y. (2013). An efficient augmented lagrangian method with applications to total variation minimization. *Computational Optimization and Applications*, 56(3):507–530.

Li, Z., Chen, Y., and Sommer, F. T. (2019). Annealed denoising score matching: Learning energy-based models in high-dimensional spaces. *arXiv preprint arXiv:1910.07762.*

Lindeberg, T. (1994). Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proc Int'l Conference on Computer Vision (ICCV).*

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver.

Lyu, S. and Simoncelli, E. P. (2009). Modeling multiscale subbands of photographic images with fields of Gaussian scale mixtures. *IEEE Trans Pattern Analysis and Machine Intelligence*, 31(4):693–706.

Ma, K., Liu, W., Zhang, K., Duanmu, Z., Wang, Z., and Zuo, W. (2018). End-to-end blind image quality assessment using deep neural networks. *IEEE Transactions on Image Processing*, 27(3):1202–1213.

Malfait, M. and Roose, D. (1997). Wavelet-based image denoising using a Markov random field a priori model. *IEEE Trans Image Processing*, 6(4):549–565.

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

Mallat, S. (2008). *A wavelet tour of signal processing: The sparse way*. Academic Press.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693.

Marchand, T., Ozawa, M., Biroli, G., and Mallat, S. (2022). Wavelet conditional renormalization group. *arXiv preprint arXiv:2207.04941*.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.

Mataev, G., Milanfar, P., and Elad, M. (2019). Deepred: Deep image prior powered by red. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0.

Meinhardt, T., Moeller, M., Hazirbas, C., and Cremers, D. (2017). Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1799–1808.

Meyes, R., Lu, M., de Puiseau, C. W., and Meisen, T. (2019). Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*.

Mihçak, M. K., Kozintsev, I., Ramchandran, K., and Moulin, P. (1999). Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Trans Signal Processing*, 6(12):300–303.

Milanfar, P. (2012a). A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128.

Milanfar, P. (2012b). A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128.

Miyasawa, K. (1961). An empirical Bayes estimator of the mean of a normal population. *Bull. Inst. Internat. Statist.*, 38:181–188.

Mohan*, S., Kadkhodaie*, Z., Simoncelli, E. P., and Fernandez-Granda, C. (2020). Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *Int'l. Conf. on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Rec.*, 65:211–222.

Paget, R. and Longstaff, I. D. (1998). Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE transactions on image processing*, 7(6):925–931.

Pang, T., Quan, Y., and Ji, H. (2020). Self-supervised bayesian deep learning for image recovery with applications to compressive sensing.

Peyré, G. and Mallat, S. (2008). Orthogonal bandlet bases for geometric images approximation. *Comm. on Pure and Applied Math.*, 61(9):1173–1212.

Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P. (2003a). Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans Image Proc*, 12(11):1338–1351. Recipient, IEEE Signal Processing Society Best Paper Award, 2008.

Portilla, J., Strela, V., Wainwright, M. J., and Simoncelli, E. P. (2003b). Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Processing*, 12(11).

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv e-prints arXiv:2204.06125.*

Raphan, M. and Simoncelli, E. P. (2011). Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420.

Reehorst, E. T. and Schniter, P. (2019). Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5(1):52–67.

Robbins, H. (1956a). An empirical Bayes approach to statistics. *Proc. Third Berkley Symposium on Mathematcal Statistics*, 1:157–163.

Robbins, H. (1956b). 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.

Romano, Y., Elad, M., and Milanfar, P. (2017a). The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844.

Romano, Y., Elad, M., and Milanfar, P. (2017b). The little engine that could: Regularization by denoising (RED). *CoRR*, abs/1611.02862.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proc IEEE/CVF Conf Computer Vision and Pattern Recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Ronneberger, O., Fischer, P., and Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. In *Int'l Conf Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer.

Rosenfeld, A., Zemel, R., and Tsotsos, J. K. (2018). The elephant in the room. *arXiv preprint arXiv:1808.03305*.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.

Saremi, S. and Hyvarinen, A. (2019). Neural empirical Bayes. *Journal of Machine Learning Research*, 20:1–23.

Saremi, S., Schölkopf, B., Mehrjou, A., and Hyvärinen, A. (2018). Deep energy estimator networks. *ArXiv e-prints (arXiv.org)*, 1805.08306.

Schmidt, U. and Roth, S. (2014). Shrinkage fields for effective image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781.

Sherrington, D. and Kirkpatrick, S. (1975). Solvable Model of a Spin-Glass. *Phys. Rev. Lett.*, 35(26):1792–1796.

Simoncelli, E. P. and Adelson, E. H. (1996). Noise removal via bayesian wavelet coring. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 1, pages 379–382. IEEE.

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

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D., editors, *Proc 32nd Int'l Conf on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France. PMLR.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. (2023). Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021a). Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428.

Song, Y. and Ermon, S. (2019a). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32*, pages 11918–11930. Curran Associates, Inc.

Song, Y. and Ermon, S. (2019b). Generative modeling by estimating gradients of the data distribution. *Adv Neural Information Processing Systems (NeurIPS)*, 32.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations. In *Int'l Conf on Learning Representations (ICLR)*.

Sun, Y., Wohlberg, B., and Kamilov, U. S. (2019). An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Teodoro, A. M., Bioucas-Dias, J. M., and Figueiredo, M. A. (2019). Image restoration and reconstruction using targeted plug-and-play priors. *IEEE Transactions on Computational Imaging*, 5(4):675–686.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *ICCV*, number 1.

Tropp, J. A. (2006). Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE transactions on information theory*, 52(3):1030–1051.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2020). Deep image prior. *Proc Int'l J Computer Vision*, pages 1867–1888.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE.

Vincent, P. (2011a). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

Vincent, P. (2011b). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Wainwright, M. J., Simoncelli, E. P., and Willsky, A. S. (2000). Random cascades of gaussian scale mixtures and their use in modeling natural images with application to denoising. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 260–263. IEEE.

Wainwright, M. J., Simoncelli, E. P., and Willsky, A. S. (2001). Random cascades on wavelet trees and their use in modeling and analyzing natural imagery. *Applied and Computational Harmonic Analysis*, 11(1):89–123.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004a). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004b). Perceptual image quality assessment: From error visibility to structural similarity. *IEEE Trans Image Processing*, 13(4):600–612.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3:1–40.

Weiss, Y., Chang, H. S., and Freeman, W. T. (2007). Learning compressed sensing. In *Snowbird Learning Workshop, Allerton, CA*.

Wiener, N. (1950). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. Technology Press.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.

Wilson, K. G. (1971). Renormalization group and critical phenomena. II. Phase-space cell analysis of critical behavior. *Physical Review B*, 4(9):3184.

Xu, X., Sun, Y., Liu†, J., Wohlberg, B., and Kamilov, U. S. (2020). Provable convergence of plug-and-play priors with MMSE denoisers. *arXiv preprint arXiv:2005.07685v1*.

Yoon, T., Choi, J. Y., Kwon, S., and Ryu, E. K. (2023). Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683.

Zhang, H., Zhou, J., Lu, Y., Guo, M., Shen, L., and Qu, Q. (2023). The emergence of reproducibility and consistency in diffusion models. *arXiv preprint arXiv:2310.05264*.

Zhang, J. and Ghanem, B. (2018). Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017a). Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Processing*, 26(7):3142–3155.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017b). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

Zhang, K., Zuo, W., Gu, S., and Zhang, L. (2017c). Learning deep cnn denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2.

Zhang, L.-Q., Cottaris, N. P., and Brainard, D. H. (2022a). An image reconstruction framework for characterizing initial visual encoding. *Elife*, 11:e71132.

Zhang, L.-Q., Kadkhodaie, Z., Simoncelli, E. P., and Brainard, D. H. (2022b). Image reconstruction from cone excitations using the implicit prior in a denoiser. *Journal of Vision*, 22(14):3793–3793.

Zhang, L.-Q., Kadkhodaie, Z., Simoncelli, E. P., and Brainard, D. H. (2024). Optimized linear measurements for inverse problems using diffusion-based image generation. *arXiv preprint arXiv:2405.17456*.

Zhang, X., Lu, Y., Liu, J., and Dong, B. (2018a). Dynamically unfolding recurrent restorer: A moving endpoint control method for image restoration. *arXiv preprint arXiv:1805.07709*.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. (2018b). Residual dense network for image restoration. *CoRR*, abs/1812.10477.