STATISTICAL SIGNAL PROCESSING APPROACHES FOR MULTI-REFERENCE ALIGNMENT AND NEURAL TEXTURE SYNTHESIS

By

Liping Yin

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Applied Mathematics—Doctor of Philosophy
Computational Mathematics, Science, and Engineering—Dual Major

2023

ABSTRACT

Statistical signal processing plays a crucial role in numerous fields of modern technology and science. Some of the important applications include extracting signals from noisy data, processing images and videos for tasks like compression and enhancement, and analyzing time-varying data, such as climate data and asset prices. In this dissertation, we address two problems related to statistical signal/image processing.

The first issue involves a generalized version of the multi-reference alignment problem in one dimension, inspired by modern data applications such as cryo-electron microscopy. The objective is to recover an unknown signal $f : \mathbb{R} \to \mathbb{R}$ from multiple observations that have been translated, dilated, and corrupted by additive noise. In the presence of large dilations and corruptions, observational data do not resemble the underlying signal. Although current approaches in the field have shown empirical success in the absence of dilations, no approach has successfully provided convergence guarantees for signal inversion while dilating, translating, and corrupting observational data all at once. Thus, we propose an unbiased estimator for the bispectrum of the unknown signal which depends on the corrupted samples and knowledge of the dilation distribution. To validate our proposed estimator, we use it for bispectrum recovery, and invert the recovered bispectrum to achieve full signal inversion.

The second problem concerns neural texture synthesis, which is important for understanding how humans perceive texture. Current approaches require regularization terms or some type of supervision to capture long range constraints, such as the alignment of bricks, in images. To remedy this issue, we propose a new set of statistics for examplar-based neural texture synthesis based on Sliced Wasserstein Loss, and augment our proposed algorithm with a multi-scale synthesis process. Based on qualitative and quantitative experiments, our results are comparable or better than current state of the art methods.

ACKNOWLEDGEMENTS

I complete the thesis thanks to the guidance of my committee members, and support from family and friends.

First of all, I would like to express my appreciation to both of my committee chairs Dr. Di Liu and Dr. Yuying Xie for supporting me during my PhD studies. It is my greatest luck to meet them in my life. I am very grateful to Dr. Yuying Xie to the point that I wish I could meet him much earlier. He offered incredible support in my hardest period of PhD life. He gave me the freedom to create my research plan and at the same time offered help whenever I need. He also help with every little thing, including checking over my thesis, applying for fellowships, and much more. I still remember he modified my slides word by word and checked my thesis writing carefully. I am impressed by his knowledge and intelligence. The most important thing is that, I am also learning from his sincerity, sense of responsibility, and perseverance. Dr. Di Liu has been my committee chair for three years. He has been very supportive and helpful during those years. He was always there when I need help for anything. I appreciate that he could give me suggestions when I was having trouble and encouraging me when I was under stress. He is also kind and wise. I would not be able to continue without his support and help.

I also very appreciate my committee members Dr. Yingda Cheng and Dr. Ekaterina Rapinchuk. There are always reachable and helpful. They went through my work and gave me helpful feedback. Their suggestions were very useful to me and I made improvement to my research. They asked helpful questions to ensure my research was sound.

I would also like to thank Dr. Mathew Hirn who has been my advisor during my third and fourth PhD life. He introduced both of my projects to me and taught me the background and all the useful math tools towards my projects. He also funded me for three semesters so that I could focus on courses and research. I would not be able to grow up without his supervision. He is very creative and able to help me when I got stuck. I also took two courses with him and I learned a lot from his courses. I also appreciated that he introduced my another important committee member, Dr. Anna Little, to me. I would like to send my best wishes to him.

As I mentioned above. Dr. Anna Little played a critical rule in my dissertation. I would not be able to finish the MRA project, which is the most important part of my dissertation, without her. She has done pioneering work on dilation MRA and guided me through the project. I am grateful that she was very patient and answered all my questions, no matter how simple they are. She read my dissertation and commented on it so that I could make improvements to it, and I'm very thankful for that already. Regardless of time zone difference she was always there to discuss with me. I still remember she checked over my proofs, came up with ideas, and debugged my code. There are too many things to thank her for. I appreciate her time and effort, and I learned a lot from her. I not only learned math techniques, but I also learned how to think scientifically. She is the great mathematician that I have learned from.

Lastly, I would like to thank my parents. Thanks for my mother always encouraging me and supporting me. And thanks for my father who has left the world along long time ago. The good memories and love will never fade.

TABLE OF CONTENTS

CHAPTER 1

THESIS OVERVIEW

We provide a brief outline of this thesis, which focuses on two problems in statistical signal processing: multi-reference alignment and texture synthesis. The second chapter focuses on a variation of the multi-reference alignment problem, which is motivated by Cryoelectron microscopy that won the Nobel prize in 2015. The original multi-reference alignment problem focuses on recovering a function $f : \mathbb{R}^3 \to \mathbb{R}$ from observations that have been translated, rotated, or corrupted by additive noise. The challenge in this problem is the following: in the presence of large corruptions, observational data do not resemble the underlying signal.

However, the multi-reference alignment problem is not necessarily an accurate representation of reality. One expects small physical variations in objects, such as macro-molecular structure in the context of Cryo-EM or deformation of an object in the context of object registration. Thus, it is of interest to consider a model where each observation has been deformed by a diffeomorphism $\tau \in C^2(\mathbb{R}^3)$. However, the class of diffeomorphisms is challenging to study. We address a simplified one-dimensional version of diffeomorphism MRA where we only consider a specific class of diffeomorphisms, dilations of the form $\tau(x) = (1 - c)x$, which we will call dilation MRA. Under mild assumptions, we propose an algorithm that is able to recover the ground truth signal using Fourier invariants, namely the power spectrum and bispectrum. Previous results provide a nonlinear estimator for bispectrum recovery, and we recover the bispectrum via a novel nonlinear estimator. This is a first step towards a model that can address general classes of diffeomorphisms, which will be left for future work.

The third chapter considers the problem of texture synthesis, which focuses on generating a new texture from a reference texture. The generated texture should have the same perceptual qualities as the reference texture without being a repetition of the old texture. We consider a modification the Heeger Bergen texture synthesis algoritm, which generates new textures via matching wavelet coefficients between a reference texture and white noise. To increase the robustness of the algorithm, we first create a nonlinear version of the wavelet transform by applying a set of nonlinearities to

each wavelet coefficient after performing a wavelet transform. We then extend our formulation by creating a modification of Mallat's invariant scattering transform, the Two Layer Scattering Pyramid. We attempt to histogram match coefficients of the Two Layer Scattering Pyramid to get good synthesis results. However, we find that our results are not competitive with state of the art algorithms, which motivates the last main chapter of this dissertation.

For chapter four, instead of using an understood and mathematically tested representation like the scattering transform, we switch to using statistics of a deep convolutional neural network, VGG19. We propose a new set of statistics for texture synthesis using Sliced Wasserstein Loss, which is an approximate solution to an optimal transport problem. Our results are compared with current state-of-the-art algorithms in single texture synthesis that have comparable run-time. We find our results are competitive with the state-of-the-art, but do require hyperparameter tuning or user-added spatial constraints. Additionally, we propose a multi-scale algorithm to augment our results, which improves the results of our synthesis dramatically.

CHAPTER 2

BISPECTRUM INVERSION FOR MULTI-REFERENCE ALIGNMENT

2.1 Introduction to MRA

The multi-reference alignment (MRA) problem, is a problem where we want to recover an unknown signal $f : \mathbb{R} \to \mathbb{R}$ from many observations that have been randomly translated and corrupted by an additive noise. Is there a way to recover f from these observations? A formal description of the assumptions is given in Model 1.

Model 1. Suppose we have M independent observations of a function $f \in L^2(\mathbb{R})$ defined by

$$f_j(x) = f(x - t_j) + \varepsilon_j(x), \quad 1 \leq j \leq M,$$

where

- $\text{supp}(f_j) \subset [-\frac{1}{2}, \frac{1}{2}]$ for $1 \leq j \leq M$.
- $\{t_j\}_{j=1}^M$ are independent samples of a random variable $t \in \mathbb{R}$.
- $\{\varepsilon_j(x)\}_{j=1}^M$ are independent white noise processes on $[-\frac{1}{2}, \frac{1}{2}]$ with variance σ^2.

The MRA problem is a simplified version of problems in cryo-electron microscopy (cryo-EM), and is similar to problems in other fields such as structural biology [17, 35, 36, 41, 42, 50], image registration [11, 18], and image processing [54].

To solve the problem outlined in Model 1, current methods can be grouped into two categories. The first of which is synchronization methods [2, 3, 4, 5, 9, 12, 13, 38, 46, 53], which try to recover each translation factor $\{t_j\}_{j=1}^M$, align each of the signals using the recovered translation factor, and average the aligned signals to get a smoother estimate for the ground truth signal. However, synchronization methods can be problematic when the signal-to-noise ratio (SNR) is small. To illustrate this point, consider the following picture:

Looking at Figure 2.1, with small amounts of noise, one can align the signals and then average the translated signals to get a cleaner version of the ground truth signal, up to a translation. However, at high noise levels, these peaks aren't recognizable, which makes this "synchronization" process unreliable.

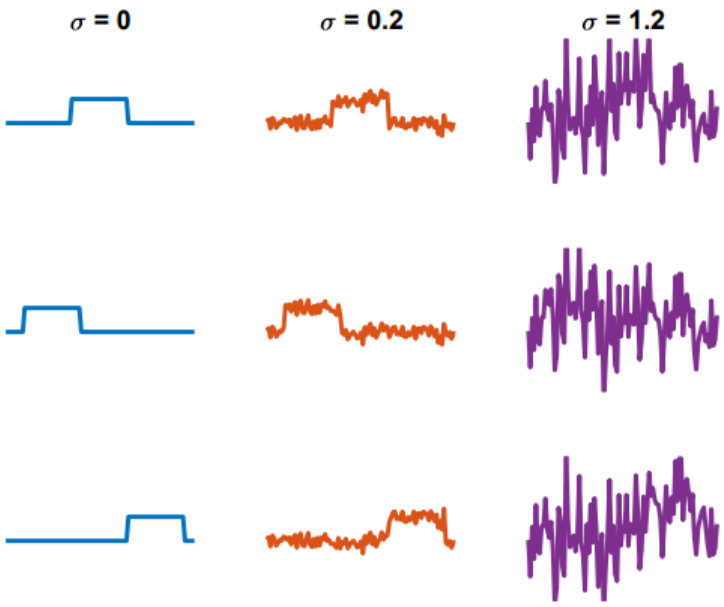


Figure 2.1 **Left Column:** three ground truth signals that have been translated without any additive noise. **Middle Column:** Adding Gaussian noise with mean zero and $\sigma^2 = 0.2$ to each of the signals in the left column. **Right Column:** Adding Gaussian noise with mean zero and $\sigma^2 = 1.2$ to each of the signals in the left column. Reference: [6].

The second approach involves estimating the signal directly using ideas such as the method of moments [22, 28, 44]; a subclass of the method of moments technique, is the method of invariants technique [2, 7, 15, 26, 27]. Additionally, expectation maximization (EM) algorithms [1, 16] have also shown success for signal recovery.

One problem with Model 1 is that it is not an accurate representation of real world applications. In particular, for three dimensional applications such as cryo-EM, molecules are randomly rotated and one only has access to 2D projections of the molecule like below:

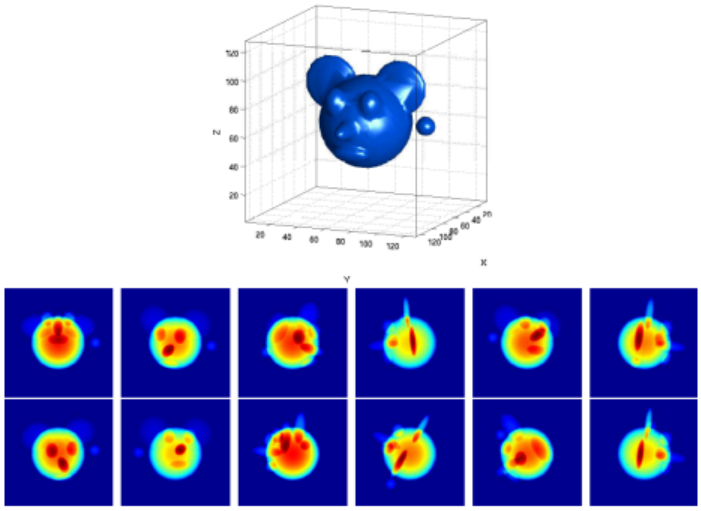


Figure 2.2 An example of the Cryo-EM problem. The 3D object at the top of the figure is the ground truth object. One has 2D slices of the object that are noisy, rotated, and possibly translated. Reference: [47].

However, the model does not consider of physical variations in the objects, such as macro-molecular structure which “flop" around. One can think of these deformations as a diffeomorphism acting on the molecule before retrieving the slices. In other words, if g is a function extracting the slices and x is a molecule, we retrieve $g(\xi(x))$, where $\xi \in C^2(\mathbb{R}^n)$ has a bijective, continuous hessian. The problem of adding random diffeomorphisms to Model 1 is much harder though. This is because the class of all diffeomorphisms encompasses a variety of different possible deformations. To simplify the problem, we consider a simple subset of all possible diffeomorphisms, the set of dilations. This leads to the formulation of Model 2 below:

Model 2. Suppose we have M independent observations of a function $f \in L^2(\mathbb{R})$ defined by

$$y_j(x) = f((1-\tau_j)^{-1}(x-t_j)) + \varepsilon_j(x) := f_j(x) + \varepsilon_j(x), \quad 1 \le j \le M$$

Furthermore, assume that

- $\operatorname{supp}(f_j) \subset [-\frac{1}{2}, \frac{1}{2}]$ for $1 \le j \le M$.
- $\{t_j\}_{j=1}^M$ are independent samples of a random variable $t \in \mathbb{R}$.
- $\{\tau_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\tau \in \mathbb{R}$ satisfying $\mathbb{E}[\tau] = 0$ and $\operatorname{Var}(\tau) = \eta^2 \le \frac{1}{12}$.

- $\{\varepsilon_j(x)\}_{j=1}^{M}$ are independent white noise processes on $[-\frac{1}{2}, \frac{1}{2}]$ with variance σ^2.

Dilations are one of the simplest class of diffeomorphisms other than translations, but Model 2 is significantly more difficult to solve compared to Model 1, even at low noise levels. Consider Figure 2.3. Intuitively, as we see in Figure 2.3, dilations add another degree of difficulty to the

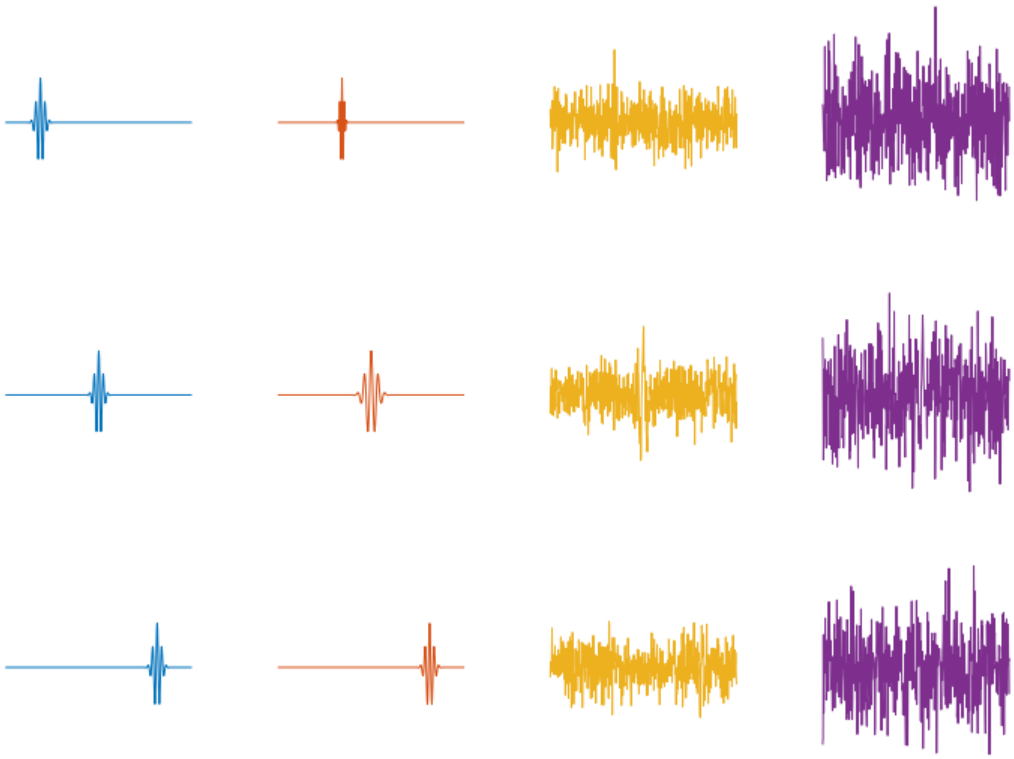


Figure 2.3 **Left Column:** three ground truth signals that have been translated without any additive noise. **Middle Left Column:** Dilating each of the signals. **Middle Right Column:** Adding Gaussian noise with mean zero and $\sigma^2 = 0.5$ to each of the signals in the left column. **Right Column:** Adding Gaussian noise with mean zero and $\sigma^2 = 2$ to each of the signals in the left column. Reference: [26].

problem. Regarding alignment, the main challenge arises from the additive noise.

Regarding the method of invariants, it utilizes translation invariant Fourier features, such as the power spectrum and bispectrum. Before we begin, define $L^q(\mathbb{R})$ as the set of functions f such that $\|f\|_q^q = \int_{\mathbb{R}} |f|^q \, dx < \infty$. The Fourier Transform of $f \in L^1(\mathbb{R})$ is

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t) e^{-iwt} \, dt, \tag{2.1}$$

the power spectrum is

$$(Pf)(\omega) = |\hat{f}(\omega)|^2, \tag{2.2}$$

and the bispectrum is

$$Bf(\omega_1, \omega_2) = \hat{f}(\omega_1)\hat{f}^*(\omega_2)\hat{f}(\omega_2 - \omega_1). \tag{2.3}$$

However, in the case of Model 2, using an approach with Fourier invariants results in significant difficulties because the Fourier Transform modulus is unstable to small dilations. Consider the operator $L_c f(t) = f((1-c)t)$. For small c, this is a minor rescaling of the function f. Let $\xi(t) = (1-c)t$; it is clear that ξ is a diffeomorphism, specifically a dilation. Then $L_c f(t) = f(\xi(t))$ with $\|\xi'\|_\infty = 1 - c$. Choose $f(t) = e^{i\alpha t}\theta(t)$, where θ is smooth with fast decay. We see that $L_c f(t) = e^{i\alpha(1-c)t}\theta((1-c)t)$, and it follows that

$$\begin{aligned}
\widehat{L_c f}(\omega) &= \int_{\mathbb{R}} \theta((1-c)t) e^{i\alpha(1-c)t - i\omega t}\, dt \\
&= \frac{1}{1-c} \int_{\mathbb{R}} \theta(t) e^{i\alpha t - i\frac{\omega}{1-c}t}\, dt \\
&= \frac{1}{1-c} \hat{\theta}((1-c)^{-1}\omega - \alpha).
\end{aligned}$$

Looking at the norm now, for $c < 1/2$, we see that from an argument identical to [34],

$$\| |\widehat{L_c f}| - |\hat{f}| \|_2^2 \approx |c\alpha| \cdot \|f\|_2^2. \tag{2.4}$$

Since α is arbitrary, we see why Fourier invariants are unstable to small dilations.

While we will address Model 2 later in this thesis, we first address a noiseless model, Model 3, given by

Model 3. Suppose we have M independent observations of a function $f \in L^2(\mathbb{R})$ defined by

$$f_j(x) = f((1-\tau_j)^{-1}(x - t_j)), \quad 1 \le j \le M$$

Furthermore, assume that

- $\operatorname{supp}(f_j) \subset [-\frac{1}{2}, \frac{1}{2}]$ for $1 \le j \le M$.
- $\{t_j\}_{j=1}^M$ are independent samples of a random variable $t \in \mathbb{R}$.

- $\{\tau_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\tau \in \mathbb{R}$ with $\mathbb{E}[\tau] = 0$ and $\text{Var}(\tau) = \eta^2 \leq \frac{1}{12}$.

Note that the lack of additive noise means one can estimate $C_f = \|f\|_2$ and then dilate each observation to have norm C_f, which makes this problem trivial. However we explore a solution to Model 3 via Fourier invariants to provide intuition on how to approach Model 2.

2.2 Notation

Let $g = Bf$. For the second and third models, let

$$g_\eta(\omega_1, \omega_2) = \mathbb{E}_\tau[(Bf_j)(\omega_1, \omega_2)].$$

We will also define the following constants and operations used in the rest of the paper:

$$C_0 = \frac{(1 - \sqrt{3}\eta)}{(1 + \sqrt{3}\eta)}\,,\ C_1 = 2\sqrt{3}\eta\,,\ C_2 = \frac{1}{1 + \sqrt{3}\eta}\,. \tag{2.5}$$

Additionally, define the dilation operator $L_C g(\omega_1, \omega_2) := C^4 g(C\omega_1, C\omega_2)$. Note that in polar coordinates (r, θ), we have $L_C g(r, \theta) = C^4 g(Cr, \theta)$.

2.3 Power Spectrum Recovery for Model 3

Specific work from [26] has produced results for recovery of the power spectrum of a hidden signal under certain assumptions in Models 2 and 3. We summarize their results below for Model 3 and explain how to adjust the results for Model 2.

Define

$$p_\eta(\omega) := \mathbb{E}_\tau\left[(Pf_j)(\omega)\right] \tag{2.6}$$

We also let $(S_C g)(\omega) = C^3 g(C\omega)$ be a rescaled dilation operator. In the case of the infinite sample limit, we have the following theorem:

Theorem 1. Assume $Pf \in \mathbf{C}^0(\mathbb{R})$ and p_η as defined in (2.6). Then for $\omega \neq 0$:

$$(Pf)(\omega) = (I - S_{C_0})^{-1} C_1 L_{C_2}(3p_\eta(\omega) + \omega p'_\eta(\omega))\,,$$

where C_0, C_1, C_2 are as defined in (2.5).

One can now use this to derive an approximation of the infinite sample estimator. In practice, one can estimate p_η by

$$\widetilde{p}_\eta(\omega) := \frac{1}{M}\sum_{j=1}^{M}(Pf_j)(\omega)\,. \tag{2.7}$$

Additionally, a natural approximation for the estimator in Theorem 1 is

$$(\widetilde{Pf})(\omega) := (I - S_{C_0})^{-1}C_1 S_{C_2}(3\widetilde{p}_\eta(\omega) + \omega\widetilde{p}'_\eta(\omega))\,, \tag{2.8}$$

For the next theorem, define the quantity:

$$(\overline{Pf})^{(k)}(\omega) := \max_{\xi\in[\omega/2,2\omega]} |(Pf)^{(k)}(\xi)| \tag{2.9}$$

Using (2.8), the following convergence theorem holds:

Theorem 2. Assume Model 3, the estimator $(\widetilde{Pf})(\omega)$ defined in (2.8), $Pf \in \mathbf{C}^3(\mathbb{R})$, and that $\omega^k(\overline{Pf})^{(k)}(\omega) \in L^2(\mathbb{R})$ for $k = 2, 3$. Then:

$$\begin{aligned}\mathbb{E}\left[\|Pf - \widetilde{Pf}\|_2^2\right] \lesssim \frac{\eta^2}{M}\big(&\|(Pf)(\omega)\|2^2 + \|\omega(Pf)'(\omega)\|_2^2 \\ &+ \|\omega^2(Pf)''(\omega)\|_2^2\big) + r\,,\end{aligned}$$

where r is a higher-order term satisfying

$$r \le \frac{\eta^4}{M}\left(\|\omega^2(\overline{Pf})''(\omega)\|_2^2 + \|\omega^3(\overline{Pf})'''(\omega)\|_2^2\right)\,.$$

That is, in expectation, the convergence rate of (2.8) to the true power spectrum is $O\left(\frac{\eta^2}{M}\right)$.

2.4 Power Spectrum Recovery for Model 2

Using these results from Model 3, we design estimators for Model 2. The main difficulty in model 2 is because of the additive noise. First of all, the mean-squared error (MSE) is only finite on a bounded interval due to the additive noise. Thus we consider a bounded frequency domain Ω, and consider the MSE of an estimator $\widetilde{Pf}$ over Ω: $\mathbb{E}\left[\|Pf - \widetilde{Pf}\|_{\mathbb{L}^2(\Omega)\|^2}\right]$.

Another difficultly is that one cannot use $\widetilde{p}_\eta$, the average of the observations without noise. Instead, one has access to

$$\frac{1}{M}\sum_{j=1}^{M} Py_j - \sigma^2 = \widetilde{p}_\eta + \widetilde{p}_\sigma$$

where

$$\widetilde{p}_\sigma := \frac{1}{M}\sum_{j=1}^{M} \widehat{f}_j\widehat{\epsilon}_j^* + \widehat{f}_j^*\widehat{\epsilon}_j + P\epsilon_j - \sigma^2 .$$

The particular problem is the term $\widetilde{p}_\sigma$, which is not continuous due to the additive noise. To remedy this issue, we smooth the noisy power spectra via a low pass filtering:

$$(\widetilde{p}_\eta + \widetilde{p}_\sigma) * \phi_L \tag{2.10}$$

using a Gaussian filter with width L, $\phi_L(\omega) = (2\pi L^2)^{-\frac{1}{2}} e^{-\frac{\omega^2}{2L^2}}$. Accordingly, we define a modified version of (2.7):

$$(\widetilde{Pf})(\omega) := (I - S_{C_0})^{-1} C_1 S_{C_2} \left[3(\widetilde{p}_\eta + \widetilde{p}_\sigma) * \phi_L(\omega) + \omega \left((\widetilde{p}_\eta + \widetilde{p}_\sigma) * \phi_L\right)'(\omega)\right] . \tag{2.11}$$

Similar to before, (2.11) is an unbiased estimator of Pf when $|M| \to \infty$ and $L \to 0$. Additionally, we have the following result:

Theorem 3. Assume Model 2, the estimator $(\widetilde{Pf})(\omega)$ defined in (2.11), $Pf \in \mathbf{C}^3(\mathbb{R})$, and that $\omega^k(\overline{Pf})^{(k)}(\omega) \in L^2(\mathbb{R})$ for $k = 2, 3$. Then

$$\mathbb{E}\left[\|Pf - \widetilde{Pf}\|^2_{L^2(\Omega)}\right] \lesssim C_{f,\Omega}\left(\frac{\eta^2}{M} + L^4 + \frac{\sigma^2 \vee \sigma^4}{L^2 M}\right) .$$

When $\sigma \geq 1$, we may choose $L = \left(\frac{\sigma^4}{M}\right)^{1/6}$. Then we have

$$\mathbb{E}\left[\|Pf - \widetilde{Pf}\|^2_{L^2(\Omega)}\right] \lesssim C_{f,\Omega}\left[\frac{\eta^2}{M} + L^4 + \left(\frac{\sigma^4}{M}\right)^{2/3}\right] .$$

We will generalize these results for bispectrum recovery in the next sections. In other words, for proper choice of L and M, the expected MSE converges to 0 as $M \to \infty$.

2.5 Bispectrum Recovery for Model 3

Following [26], we propose a similar process for creating an estimator for the bispectrum. We will first consider the case where we have an infinite number of samples and find an unbiased estimator.

Theorem 4. Suppose we have an infinite number of samples and assume that $Bf \in C^1(\mathbb{R}^2)$ An unbiased estimator for the bispectrum is given by

$$Bf(r,\theta) = (I - L_{C_0})^{-1} C_1 L_{C_2} \left(4g_\eta(r,\theta) + r\frac{\partial g_\eta}{\partial r}(r,\theta) \right).$$

Proof. Recall that the Bispectrum is given by

$$Bf(\omega_1,\omega_2) = \hat{f}(\omega_1)\hat{f}^*(\omega_2)\hat{f}(\omega_2 - \omega_1).$$

The Fourier Transform of each f_j is $e^{-i\omega t_j}(1-\tau_j)\hat{f}((1-\tau_j)\omega)$. Making a substitution,

$$\begin{aligned} Bf_j(\omega_1,\omega_2) &= (1-\tau_j)^3 e^{-i\omega_1 t_j}\hat{f}((1-\tau_j)\omega_1) \\ &\quad \cdot [e^{-i\omega_2 t_j}\hat{f}((1-\tau_j)\omega_1)]^* \cdot e^{-i(\omega_2-\omega_1)t_j}\hat{f}((1-\tau_j)(\omega_2-\omega_1)) \\ &= (1-\tau_j)^3 \hat{f}((1-\tau_j)\omega_1)\hat{f}^*((1-\tau_j)\omega_1)\hat{f}((1-\tau_j)(\omega_2-\omega_1)). \end{aligned}$$

So

$$(Bf_j)(\omega_1,\omega_2) = (1-\tau_j)^3 (Bf)((1-\tau_j)\omega_1,(1-\tau_j)\omega_2).$$

Since τ has uniform distribution with variance η^2, the pdf of τ has form $p_\tau = \frac{1}{2\sqrt{3}\eta}\chi_{[-\sqrt{3}\eta,\sqrt{3}\eta]}$. Thus:

$$\begin{aligned} g_\eta(\omega_1,\omega_2) &= \mathbb{E}_\tau[Bf_j(\omega_1,\omega_2)] \\ &= \mathbb{E}_\tau[(1-\tau)^3 g((1-\tau)w_1,(1-\tau)w_2)] \\ &= \int (1-\tau)^3 g((1-\tau)w_1,(1-\tau)w_2)p_\tau(\tau)\,d\tau \end{aligned}$$

Now we convert to polar coordinates (r,θ) and let $\tilde{\tau} = (1-\tau)r$:

$$\begin{aligned} g_\eta(r,\theta) &= \frac{1}{2\sqrt{3}\eta}\int_{-\sqrt{3}\eta}^{\sqrt{3}\eta}(1-\tau)^3 g((1-\tau)r,\theta)d\tau \\ &= \frac{1}{2\sqrt{3}\eta r^4}\int_{(1-\sqrt{3}\eta)r}^{(1+\sqrt{3}\eta)r}\tilde{\tau}^3 g(\tilde{\tau},\theta)\,d\tilde{\tau}. \end{aligned}$$

Let H be the antiderivative in the variable w for the function $h(w,\theta) = w^3 g(w,\theta)$. In other words,

$$\frac{\partial H}{\partial w}(w,\theta) = h(w,\theta) = w^3 g(w,\theta).$$

By Fundamental Theorem of Calculus,

$$2\sqrt{3}\eta r^4 g_\eta(r,\theta) = H((1+3\sqrt{\eta})r,\theta) - H((1-3\sqrt{\eta})r,\theta).$$

Now take derivative with respect to r and divide both sides by r^3 to get

$$2\sqrt{3}\eta\left(4g_\eta(r,\theta) + r\frac{\partial g_\eta}{\partial r}(r,\theta)\right) = (1+3\sqrt{\eta})^4 g((1+3\sqrt{\eta})r,\theta) - (1-3\sqrt{\eta})^4 g((1-3\sqrt{\eta})r,\theta).$$

We now apply the dilation operation L_{C_2} to both sides, which yields

$$C_1 L_{C_2}\left(4g_\eta(r,\theta) + r\frac{\partial g_\eta}{\partial r}(r,\theta)\right) = g(r,\theta) - \left(\frac{1-3\sqrt{\eta}}{1+3\sqrt{\eta}}\right)^4 g\left(\frac{1-3\sqrt{\eta}}{1+3\sqrt{\eta}}r,\theta\right).$$

We can also rewrite the right side in terms of I and L_{C_0} to get

$$C_1 L_{C_2}\left(4g_\eta(r,\theta) + r\frac{\partial g_\eta}{\partial r}(r,\theta)\right) = (I - L_{C_0})g(r,\theta).$$

Thus, an unbiased estimator is

$$g(r,\theta) = (I - L_{C_0})^{-1} C_1 L_{C_2}\left(4g_\eta(r,\theta) + r\frac{\partial g_\eta}{\partial r}(r,\theta)\right).$$

□

However, we are only given a finite number of samples and do not have access to the estimator above in actual applications. For Model 3, we can approximate g_η by taking an average of M samples using

$$\widetilde{g}_\eta(\omega_1,\omega_2) := \frac{1}{M}\sum_{j=1}^{M}(Bf_j)(\omega_1,\omega_2).$$

Based on Proposition 4, a good choice for the estimator is

$$(\widetilde{Bf})(r,\theta) := (I - L_{C_0})^{-1} C_1 L_{C_2}\left(4\tilde{g}_\eta(r,\theta) + r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta)\right).$$

To show the estimator $\widetilde{Bf}$ has a small error, we will need the following lemma.

Lemma 1. Assume that $Bf \in C^1(\mathbb{R})$. Then

$$\|Bf(r,\theta) - \widetilde{Bf}(r,\theta)\|_2^2 \lesssim \|g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta)\|_2^2 + \left\|r\frac{\partial g_\eta}{\partial r}(r,\theta) - r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta)\right\|_2^2.$$

Proof. We start with

$$Bf(r,\theta) - \widetilde{Bf}(r,\theta) = (I - L_{C_0})^{-1} C_1 L_{C_2} \left(4(g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta)) + r\left(\frac{\partial g_\eta}{\partial r}(r,\theta) - \frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta) \right) \right).$$

By using the triangle inequality,

$$\begin{aligned} \|Bf(r,\theta) - \widetilde{Bf}(r,\theta)\|_2^2 \leq\ & 32C_1^2 \|(I - L_{C_0})^{-1}\|^2 \|L_{C_2}\|^2 \|g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta)\|_2^2 \\ & + 2C_1^2 \|(I - L_{C_0})^{-1}\|^2 \|L_{C_2}\|^2 \left\| r\frac{\partial g_\eta}{\partial r}(r,\theta) - r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta) \right\|_2^2. \end{aligned}$$

To compute the spectral norm of L_C, we revert back to Cartesian coordinates:

$$\|L_C^m g_\eta\|_2^2 = C^{8m} \int_{\mathbb{R}} \int_{\mathbb{R}} |g_\eta(C^m\omega_1, C^m\omega_2)|^2 d\omega_1 d\omega_2.$$

Let $u = C^m\omega_1$ and $v = C^m\omega_2$:

$$\|L_C^m g_\eta\|_2^2 = C^{6m} \int_{\mathbb{R}} \int_{\mathbb{R}} |g_\eta(u,v)|^2 du dv = C^{6m} \|g_\eta\|_2^2.$$

Thus $\|L_{C_2}\|^2 = C_2^{6m}$. Our result also implies that

$$\|(I - L_{C_0})^{-1}\| \leq \sum_{j=0}^{\infty} \left\| L_{C_0}^j \right\| \leq \sum_{j=0}^{\infty} C_0^{3j} = \frac{1}{1 - C_0^3}.$$

Putting everything together,

$$\begin{aligned} \|Bf(r,\theta) - \widetilde{Bf}(r,\theta)\|_2^2 \leq\ & \frac{16 \cdot 12\eta^2}{(1 - C_0^6)^2 (1 + \sqrt{3}\eta)^{12}} \|g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta)\|_2^2 \\ & + \frac{12\eta^2}{(1 - C_0^6)^2 (1 + \sqrt{3}\eta)^{12}} \left\| r\frac{\partial g_\eta}{\partial r}(r,\theta) - r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta) \right\|_2^2. \end{aligned}$$

Now it suffices to prove that $\frac{12\eta^2}{(1-C_0^6)^2(1+\sqrt{3}\eta)^{12}}$ is bounded by constant:

$$\begin{aligned}
\frac{12\eta^2}{(1-C_0^6)^2(1+\sqrt{3}\eta)^{12}} &\leq \frac{12\eta^2}{\left[1-\left(\frac{1-\sqrt{3}\eta}{1+\sqrt{3}\eta}\right)^6\right]^2} \\
&= \frac{12\eta^2}{\left[\left(\frac{1+\sqrt{3}\eta}{1+\sqrt{3}\eta}\right)^6-\left(\frac{1-\sqrt{3}\eta}{1+\sqrt{3}\eta}\right)^6\right]^2} \\
&= \frac{12\eta^2(1+\sqrt{3}\eta)^{12}}{(108\sqrt{3}\eta^5+120\sqrt{3}\eta^3+12\sqrt{3}\eta)^2} \\
&= \frac{12\eta^2(1+\sqrt{3}\eta)^{12}}{\eta^2(108\sqrt{3}\eta^4+120\sqrt{3}\eta^2+12\sqrt{3})^2} \\
&\lesssim \frac{\eta^2}{\eta^2}(1+\sqrt{3}\eta)^{12} \\
&= O(1).
\end{aligned}$$

□

This lemma implies that we only have to bound

$$\begin{aligned}
&\mathbb{E}\left[\|g_\eta(r,\theta)-\tilde{g}_\eta(r,\theta)\|_2^2\right], \\
&\mathbb{E}\left[\left\|r\frac{\partial g_\eta}{\partial r}(r,\theta)-r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta)\right\|_2^2\right]
\end{aligned}$$

with an $O(1/M)$ bound on M for $\widetilde{Bf}(r,\theta)$ to converge to $Bf(r,\theta)$ with an $O(1/M)$ bound. We have the following result now.

Theorem 5. Suppose that $Bf \in C^3(\mathbb{R}^2)$ and also assume that $r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|^2 \in L^2(\mathbb{R}^2, dr\times d\theta)$ and $r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)|^2 \in L^2(\mathbb{R}^2, dr\times d\theta)$. Then the following bound holds:

$$\begin{aligned}
\mathbb{E}\left[\|Bf-\widetilde{Bf}\|_2^2\right] \lesssim \frac{\eta^2}{M}&\left(\|(Bf)(r,\theta)\|_2^2+2\|r\partial_r(Bf)(r,\theta)\|_2^2+\|r^2\partial_{rr}(Bf)(r,\theta)\|_2^2\right) \\
&+\frac{\eta^4}{M}\left(\left\|r^2\max_{\alpha\in[r/2,2r]}\partial_{\alpha\alpha}(Bf)(\alpha,\theta)\right\|_2^2+\left\|r^3\max_{\gamma\in[r/2,2r]}\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)\right\|_2^2\right).
\end{aligned}$$

with

$$(\widetilde{Bf})(r,\theta) = (I-L_{C_0})^{-1}C_1L_{C_2}\left(4\tilde{g}_\eta(r,\theta)+r\frac{\partial\tilde{g}_\eta}{\partial r}(r,\theta)\right).$$

Proof. First, assume that the $Bf : \mathbb{R}^2 \to \mathbb{R}$. The argument will be generalized to the complex case after. Notice that

$$|\tilde{g}_\eta(r,\theta) - g_\eta(r,\theta)|^2 = \left| \frac{1}{M} \sum_{j=1}^{M} (Bf_j)(r,\theta) - g_\eta(r,\theta) \right|^2 .$$

Define

$$X_j = Bf_j(r,\theta) - g_\eta(r,\theta) = Bf_j(r,\theta) - \mathbb{E}[(Bf_j)(r,\theta)].$$

This means each X_j is zero centered, so we have

$$\mathbb{E}\left[\left| \frac{1}{M} \sum_{j=1}^{M} X_j \right|^2 \right] = \operatorname{var}\left[\frac{1}{M} \sum_{j=1}^{M} X_j \right] = \frac{\operatorname{var}(X_j)}{M}.$$

Write

$$X_j = Bf_j(r,\theta) - (Bf)(r,\theta) + (Bf)(r,\theta) - \mathbb{E}[(Bf_j)(r,\theta)].$$

Then

$$X_j^2 \lesssim (Bf_j(r,\theta) - (Bf)(r,\theta))^2 + ((Bf)(r,\theta) - \mathbb{E}[(Bf_j)(r,\theta)])^2$$

and

$$\begin{aligned} \mathbb{E}[X_j^2] &\lesssim \mathbb{E}\left[(Bf_j(r,\theta) - (Bf)(r,\theta))^2 \right] + \mathbb{E}\left[((Bf)(r,\theta) - \mathbb{E}[(Bf_j)(r,\theta)])^2 \right] \\ &\lesssim \mathbb{E}\left[(Bf_j(r,\theta) - (Bf)(r,\theta))^2 \right]. \end{aligned}$$

Each τ_j has bounded variance and is supported on $[-1/2, 1/2]$. Taylor expand the dilated bispectrum in radial variable in interval $[r/2, 2r]$:

$$(Bf)((1-\tau_j)r,\theta) = (Bf)(r,\theta) - \partial_r(Bf)(r,\theta) r\tau_j + \frac{1}{2}\partial_{rr}(Bf)(r,\theta)\Big|_{r=\alpha} r^2\tau_j^2, \quad \alpha \in [r/2, 2r].$$

Now multiply both sides by $(1-\tau_j)^3$ to get

$$\begin{aligned} (Bf_j)(r,\theta) &= (1-\tau_j)^3 (Bf)((1-\tau_j)r,\theta) \\ &= (1-\tau_j)^3 (Bf)(r,\theta) - (1-\tau_j)^3 \partial_r(Bf)(r,\theta) r\tau_j \\ &\quad + (1-\tau_j)^3 \frac{1}{2}\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2\tau_j^2. \end{aligned}$$

with $\alpha \in [r/2, 2r]$. It now follows that

$$\begin{aligned}(Bf_j)(r,\theta) - (Bf)(r,\theta) = (3\tau_j^2 - 3\tau_j - \tau_j^3)(Bf)(r,\theta) \\ - (1-\tau_j)^3 \partial_r (Bf)(r,\theta) r\tau_j \\ + \frac{1}{2}(1-\tau_j)^3 \partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2 \tau_j^2.\end{aligned}$$

with $\alpha \in [r/2, 2r]$.

Square both sides to get:

$$\begin{aligned}((Bf_j)(r,\theta) - (Bf)(r,\theta))^2 = (3\tau_j^2 - \tau_j - \tau_j^3)^2 (Bf)^2(r,\theta) \\ - 2(3\tau_j^2 - \tau_j - \tau_j^3)(1-\tau_j)^3 (Bf)(r,\theta)\partial_r(Bf)(r,\theta) r\tau_j \\ + (3\tau_j^2 - \tau_j - \tau_j^3)(1-\tau_j)^3 (Bf)(r,\theta)\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2\tau_j^2 \\ + (1-\tau_j)^6 [\partial_r(Bf)(r,\theta)]^2 r^2 \tau_j^2 \\ - (1-\tau_j)^6 \partial_r(Bf)(r,\theta)\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^3\tau_j^3 \\ + \frac{(1-\tau_j)^6}{4}[\partial_{\alpha\alpha}(Bf)(\alpha,\theta)]^2 r^4 \tau_j^4.\end{aligned}$$

Using the inequality $2|ab| \le |a|^2 + |b|^2$, it follows that

$$\begin{aligned}2|(\tau_j^2 - \tau_j - \tau_j^3)(1-\tau_j)^3 (Bf)(r,\theta)\partial_r(Bf)(r,\theta) r\tau_j| \le (\tau_j^2 - \tau_j - \tau_j^3)^2 |(Bf)(r,\theta)|^2 \\ + (1-\tau_j)^6 |\partial_r(Bf)(r,\theta) r\tau_j|^2\end{aligned}$$

and

$$\begin{aligned}|(3\tau_j^2 - \tau_j - \tau_j^3)(1-\tau_j)^3 (Bf)(r,\theta)\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2\tau_j^2| \le \frac{1}{2}(3\tau_j^2 - \tau_j - \tau_j^3)^2 |(Bf)(r,\theta)|^2 \\ + \frac{1}{2}|1-\tau_j|^6 |\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2\tau_j^2|^2.\end{aligned}$$

and

$$\begin{aligned}|(1-\tau_j)^6 \partial_r(Bf)(r,\theta)\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^3\tau_j^3| \le \frac{1}{2}(1-\tau_j)^6 |\partial_r(Bf)(r,\theta)\tau_j r|^2 \\ + \frac{1}{2}|1-\tau_j|^6 |\partial_{\alpha\alpha}(Bf)(\alpha,\theta) r^2\tau_j^2|^2.\end{aligned}$$

Take the expectation of both sides now. Since the pdf of τ is supported on $[-1/2, 1/2]$ and zero centered,

$$\mathbb{E}[(3\tau_j^2 - \tau_j - \tau_j^3)^2] \lesssim \mathbb{E}[\tau_j^2] \lesssim \eta^2.$$

The other terms with τ_j are bounded in a similar way. Thus

$$\begin{aligned}\mathbb{E}[((Bf_j)(r,\theta) - (Bf)(r,\theta))^2] \lesssim \eta^2 (Bf)^2(r,\theta) + r^2\eta^2[\partial_r(Bf)(r,\theta)]^2 \\ + \eta^4 r^4 \left[\max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|\right]^2.\end{aligned}$$

We also have

$$\begin{aligned}\mathrm{var}[X_j] &= \mathbb{E}[X_j^2] \\ &\lesssim \mathbb{E}[((Bf_j)(r,\theta) - (Bf)(r,\theta))^2] \\ &\lesssim \eta^2 (Bf)^2(r,\theta) + r^2\eta^2[\partial_r(Bf)(r,\theta)]^2 + \eta^4 r^4 \left[\max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|\right]^2,\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left[(g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta))^2\right] \lesssim \frac{\eta^2}{M}(Bf)^2(r,\theta) + r^2\frac{\eta^2}{M}[\partial_r(Bf)(r,\theta)]^2 \\ + \frac{\eta^4}{M} r^4 \left[\max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|\right]^2.\end{aligned}$$

Now we can take the integral and expectation to get

$$\begin{aligned}\mathbb{E}\left[\|g_\eta(r,\theta) - \tilde{g}_\eta(r,\theta)\|_2^2\right] \lesssim \frac{\eta^2}{M}\|(Bf)(r,\theta)\|_2^2 + \frac{\eta^2}{M}\|r\partial_r(Bf)(r,\theta)\|_2^2 \\ + \frac{\eta^4}{M}\left\|r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|\right\|_2^2\end{aligned}$$

The first term is now handled appropriately. We can now repeat a nearly identical argument for the second term. Let $g_j = Bf_j$ and

$$Z_j = r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g_\eta}{\partial r}(r,\theta).$$

We have

$$r\frac{\partial \tilde{g}_\eta}{\partial r}(r,\theta) - r\frac{\partial g_\eta}{\partial r}(r,\theta) = \frac{1}{M}\sum_{j=1}^M r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g_\eta}{\partial r}(r,\theta).$$

By Leibniz Rule, we can take the derivative inside the expectation to get $\mathbb{E}[Z_j] = 0$, and a similar argument from before yields

$$Z_j^2 \lesssim \left[r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g}{\partial r}(r,\theta) \right]^2 + \left[r\frac{\partial g}{\partial r}(r,\theta) - r\frac{\partial g_\eta}{\partial r}(r,\theta) \right]^2$$

and

$$\mathbb{E}[Z_j^2] \lesssim \mathbb{E}\left[\left(r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g}{\partial r}(r,\theta) \right)^2 \right].$$

Taylor expand $\partial_r(Bf)((1-\tau_j)r,\theta)$ to get

$$\partial_r(Bf)((1-\tau_j)r,\theta) = \partial_r(Bf)(r,\theta) - \partial_{rr}(Bf)(r,\theta)r\tau_j + \frac{1}{2}\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)r^2\tau_j^2, \quad \gamma \in [r/2, 2r].$$

Since $r\frac{\partial g_j}{\partial r}(r,\theta) = r(1-\tau_j)^4\partial_r(Bf)((1-\tau_j)r,\theta)$, multiply both sides by $(1-\tau_j)^4$:

$$\begin{aligned}(1-\tau_j)^4 r\partial_r(Bf)((1-\tau_j)r,\theta) &= (1-\tau_j)^4 r\partial_r(Bf)(r,\theta) \\ &\quad - (1-\tau_j)^4\partial_{rr}(Bf)(r,\theta)r^2\tau_j + (1-\tau_j)^4\frac{1}{2}\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)r^3\tau_j^2\end{aligned}$$

with $\gamma \in [r/2, 2r]$. Then

$$\begin{aligned}r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g}{\partial r}(r,\theta) &= (\tau_j^4 - 4\tau_j^3 + 6\tau_j^2 - 4\tau_j)r\partial_r(Bf)(r,\theta) \\ &\quad - (1-\tau_j)^4\partial_{rr}(Bf)(r,\theta)r^2\tau_j + (1-\tau_j)^4\frac{1}{2}\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)r^3\tau_j^2\end{aligned}$$

with $\gamma \in [r/2, 2r]$. By a similar process from above,

$$\begin{aligned}\mathbb{E}\left[\left(r\frac{\partial g_j}{\partial r}(r,\theta) - r\frac{\partial g_\eta}{\partial r}(r,\theta) \right)^2 \right] &\lesssim \frac{\eta^2}{M}\|r\partial_r(Bf)(r,\theta)\|_2^2 \\ &\quad + \frac{\eta^2}{M}\|r^2\partial_{rr}(Bf)(r,\theta)\|_2^2 + \frac{\eta^4}{M}\left\| r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)| \right\|_2^2.\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}\left[\|Bf - \widetilde{Bf}\|_2^2 \right] &\lesssim \frac{\eta^2}{M}\left(\|(Bf)(r,\theta)\|_2^2 + 2\|r\partial_r(Bf)(r,\theta)\|_2^2 + +\|r^2\partial_{rr}(Bf)(r,\theta)\|_2^2 \right) \\ &\quad + \frac{\eta^4}{M}\left(\left\| r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)| \right\|_2^2 + \left\| r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)| \right\|_2^2 \right).\end{aligned}$$

For the case where $Bf : \mathbb{R}^2 \to \mathbb{C}$, simply write $Bf = \mathrm{Re}(Bf) + i\mathrm{Im}(Bf)$ and repeat the argument above. Then it follows that

$$\begin{aligned}
&\mathbb{E}\left[\|\mathrm{Re}(Bf) - \mathrm{Re}(\widetilde{Bf})\|_2^2\right] \\
&\lesssim \frac{\eta^2}{M}\left(\|\mathrm{Re}(Bf)(r,\theta)\|_2^2 + 2\|r\partial_r \mathrm{Re}(Bf)(r,\theta)\|_2^2 + +\|r^2\partial_{rr}\mathrm{Re}(Bf)(r,\theta)\|_2^2\right) \\
&+ \frac{\eta^4}{M}\left(\left\|r^2 \max_{\alpha\in[r/2,2r]} |\partial_\alpha \mathrm{Re}(Bf)(\alpha,\theta)|\right\|_2^2 + \left\|r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\alpha\alpha\alpha}\mathrm{Re}(Bf)(\gamma,\theta)|\right\|_2^2\right)
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}\left[\|\mathrm{Im}(Bf) - \mathrm{Im}(\widetilde{Bf})\|_2^2\right] \\
&\lesssim \frac{\eta^2}{M}\left(\|\mathrm{Im}(Bf)(r,\theta)\|_2^2 + 2\|r\partial_r \mathrm{Im}(Bf)(r,\theta)\|_2^2 + +\|r^2\partial_{rr}\mathrm{Im}(Bf)(r,\theta)\|_2^2\right) \\
&+ \frac{\eta^4}{M}\left(\left\|r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha} \mathrm{Im}(Bf)(\alpha,\theta)\right\|_2^2 + \left\|r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}\mathrm{Im}(Bf)(\gamma,\theta)|\right\|_2^2\right).
\end{aligned}$$

Adding the two inequalities together yields the desired result. □

Now that we have solved the noiseless case, the goal is to move onto Model 2 and try to adapt our method to work in the presence of additive noise.

2.6 Bispectrum Recovery for Model 2

Now we move on to Model 2. This is similar, but the additive noise creates two difficulties. First, we must restrict ourselves from $\mathbb{R}^2$ to some finite domain Ω since the MSE is not well defined on infinite intervals because of the noise. Second, we don't necessarily have access to $\tilde{g}_\eta$ like before. Instead, we only know

$$\frac{1}{M}\sum_{j=1}^{M} By_j.$$

Writing $\hat{\varepsilon}(\omega) = \int_{-1/2}^{1/2} e^{-i\omega x} dB_x$ as an integral with respect to a Brownian motion, it is clear that $\mathbb{E}[B\varepsilon_j] = 0$. Also, notice that $\mathbb{E}[B\varepsilon_j] = \mathbb{E}[\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2 - \omega_1)]$, where each ε_j is white

noise process on $[-1/2, 1/2]$ with variance σ^2. We see that for each index

$$\begin{aligned}
By_j(\omega_1,\omega_2) &= B(f_j+\varepsilon_j)(\omega_1,\omega_2)\\
&= (\hat{f}_j(\omega_1)+\hat{\varepsilon}_j(\omega_1))(\hat{f}_j^*(\omega_2)+\hat{\varepsilon}_j^*(\omega_2))(\hat{f}_j(\omega_2-\omega_1)+\hat{\varepsilon}_j(\omega_2-\omega_1))\\
&= \hat{f}_j(\omega_1)\hat{f}_j^*(\omega_2)\hat{f}_j(\omega_2-\omega_1)+\hat{f}_j(\omega_1)\hat{f}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1)\\
&+ \hat{f}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1)+\hat{f}_j^*(\omega_2)\hat{f}_j(\omega_2-\omega_1)\hat{\varepsilon}_j(\omega_1)\\
&+ \hat{f}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j(\omega_2-\omega_1)+\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{f}_j(\omega_2-\omega_1)\\
&+ \hat{f}_j(\omega_1)\hat{f}_j(\omega_2-\omega_1)\hat{\varepsilon}_j^*(\omega_2)+\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1)\\
&:= Bf_j(\omega_1,\omega_2)+R_j(\omega_1,\omega_2).
\end{aligned}$$

Thus, we need to perform a σ-based centering to recover $\frac{1}{M}\sum_{j=1}^{M} Bf_j(\omega_1,\omega_2)$. We see that if we take the expectation over ε, we have

$$\begin{aligned}
\mathbb{E}_\varepsilon[By_j(\omega_1,\omega_2)] &= Bf_j(\omega_1,\omega_2)+\mathbb{E}_\varepsilon[\hat{f}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1)\\
&\quad+\hat{f}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j(\omega_2-\omega_1)+\hat{\varepsilon}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{f}_j(\omega_2-\omega_1)].
\end{aligned}$$

Now we know from Theorem 4.5 of [29]

$$\mathbb{E}_\varepsilon[\hat{\varepsilon}(\omega_1)\hat{\varepsilon}^*(\omega_2)] = 2\sigma^2\frac{\sin\left(\frac{1}{2}(\omega_2-\omega_1)\right)}{\omega_2-\omega_1}.$$

Define $h(\omega) = 2\sigma^2\frac{\sin(\frac{1}{2}\omega)}{\omega}$. Note that h is an even function since it is the product of two odd functions. Taking the expectation yields

$$\mathbb{E}_\varepsilon[By_j(\omega_1,\omega_2)] = Bf_j(\omega_1,\omega_2)+\hat{f}_j(\omega_1)h(\omega_1)+\hat{f}_j^*(\omega_2)h(\omega_2)+\hat{f}_j(\omega_2-\omega_1)h(\omega_2-\omega_1).$$

Now take the expectation over the translation and dilation parameters to get

$$\begin{aligned}
\mathbb{E}[By_j(\omega_1,\omega_2)] &= \mathbb{E}[Bf_j(\omega_1,\omega_2)]+\mathbb{E}[\hat{f}_j(\omega_1)]h(\omega_1)\\
&\quad+\mathbb{E}[\hat{f}_j^*(\omega_2)]h(\omega_2)+\mathbb{E}[\hat{f}_j(\omega_2-\omega_1)]h(\omega_2-\omega_1).
\end{aligned}$$

Denote the $\mu(\omega) = \mathbb{E}[\hat{f}_j(\omega)]$ and let $\tilde{\mu}(\omega) = \frac{1}{M}\sum_{j=1}^{M} \hat{y}_j(\omega)$. We will approximate the expectation using

$$\begin{aligned}\mathbb{E}[Bf_j] &\approx \frac{1}{M}\sum_{j=1}^{M} By_j - \tilde{\mu}(\omega_1)h(\omega_1) - \tilde{\mu}^*(\omega_2)h(\omega_2) - \tilde{\mu}(\omega_2-\omega_1)h(\omega_2-\omega_1) \\ &= \frac{1}{M}\sum_{j=1}^{M} By_j - R_\sigma.\end{aligned}$$

After empirical centering by R_σ, we can thus decompose the computable quantity into two pieces:

$$\frac{1}{M}\sum_{j=1}^{M} By_j - R_\sigma = \tilde{g}_\eta + \tilde{g}_\sigma,$$

where

$$\tilde{g}_\sigma = \frac{1}{M}\sum_{j=1}^{M} R_j(\omega_1,\omega_2) - R_\sigma(\omega_1,\omega_2) = \frac{1}{M}\sum_{j=1}^{M} R_j(r,\theta) - R_\sigma(r,\theta).$$

The term $\tilde{g}_\eta + \tilde{g}_\sigma$ is not smooth due to the additive noise. Thus we need to add some procedure to smooth the signal. Let $\phi_L(r) = (2\pi L^2)^{-1/2} e^{-r^2/(2L^2)}$ be a low pass filter. We define a new estimator for Model 2 as

$$(\widetilde{Bf})(r,\theta) := (I - L_{C_0})^{-1} C_1 L_{C_2} \left(4((\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L)(r,\theta) + r\partial_r((\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L)(r,\theta)\right). \quad (2.12)$$

We will use the following two lemmas, whose proofs are similar to [26].

Lemma 2. Let $q \in L^2(\mathbb{R}^2)$ and assume $|\hat{q}(\omega)|$ decays like $|q|^{-\alpha}$ for some integer $\alpha \geq 2$ and ω such that $|\omega| \geq \omega_0$ for some $\omega_0 > 0$. Then for L small enough,

$$\|q - q * \phi_L\|_2^2 \lesssim \|q\|_2^2 L^4 + L^{4\wedge(2\alpha-2)}.$$

Proof. We have $\hat{\phi}_L(|\omega|) = e^{-L^2|\omega|^2/2}$ and

$$1 - \hat{\phi}_L(|\omega|) = \frac{L^2|\omega|^2}{2} + O(L^3).$$

Letting $|\omega| = r$, we write the integral in polar coordinates and get

$$\begin{aligned}
\|q - q * \phi_L\|_2^2 &= \frac{1}{(2\pi)^2}\|\hat{q}(1-\hat{\phi}_L)\|_2^2 \\
&= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{\infty}|\hat{q}(r,\theta)|^2|1-\hat{\phi}_L(r)|^2 r\,dr\,d\theta \\
&= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{\omega_0}|\hat{q}(r,\theta)|^2|1-\hat{\phi}_L(\hat{r})|^2 r\,dr\,d\theta \\
&+ \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_{\omega_0}^{\infty}|\hat{q}(r,\theta)|^2|1-\hat{\phi}_L(r)|^2 r\,dr\,d\theta \\
&:= I_1 + I_2.
\end{aligned}$$

For I_1, we have

$$\begin{aligned}
I_1 &= \frac{1}{(2\pi)^2}\int_0^{2\pi}\int_0^{\omega_0}|\hat{q}(r,\theta)|^2\left(\frac{L^2|r|^2}{2}+O(L^3)\right)^2 r\,dr\,d\theta \\
&\le \frac{1}{2\pi^2}\left(\frac{L^4\omega_0^4}{4}+O(L^5)\right)\int_0^{2\pi}\int_0^{\omega_0}|\hat{q}(r,\theta)|^2 r\,dr\,d\theta \\
&\lesssim \omega_0^4\|q\|_2^2 L^4 + O(L^5).
\end{aligned}$$

For I_2,

$$\begin{aligned}
I_2 &\le \frac{C^2}{(2\pi)^2}\int_0^{2\pi}\int_1^{\infty} r^{-2\alpha+1}(1-e^{-L^2r^2/2})^2\,dr\,d\theta \\
&= \frac{C^2}{2\pi}\int_1^{\infty} r^{-2\alpha+1}(1-e^{-L^2r^2/2})^2\,dr.
\end{aligned}$$

Using the same argument as [26], it follows that $I_2 \lesssim L^{4\wedge(2\alpha-2)}$. □

Lemma 3. Let $rq(r,\theta) \in L^2(\mathbb{R}^2, dr\times d\theta)$ and assume its Fourier transform $\widehat{(\cdot)q(\cdot)}(\omega)$ decays like $|w|^{-\alpha}$ for some integer $\alpha \ge 2$. Then for L small enough,

$$\|r(q - q*\phi_L)(r,\theta)\|_2^2 \lesssim \|rq(r,\theta)\|_2^2 L^4 + L^{4\wedge(2\alpha-2)} + L^3\|q\|_2^2.$$

Proof. First, we switch back to rectangular coordinates:

$$\begin{aligned}
\|r(q - q * \phi_L)(r,\theta)\|_2^2 &= \|(\omega_1^2 + \omega_2^2)^{1/2}(q - q * \phi_L)(\omega_1,\omega_2)\|_2^2 \\
&= \int_{\mathbb{R}^2} w_1^2 |(q - q * \phi_L)(\omega_1,\omega_2)|^2 \, d\omega_1 \, d\omega_2 \\
&+ \int_{\mathbb{R}^2} w_2^2 |(q - q * \phi_L)(\omega_1,\omega_2)|^2 \, d\omega_1 \, d\omega_2 \\
&= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\partial_{t_1}\hat{q}(t_1,t_2) - \partial_{t_1}(\hat{q}(t_1,t_2)\hat{\phi}_L(t_1,t_2))|^2 \, dt_1 \, dt_2 \\
&+ \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\partial_{t_2}\hat{q}(t_1,t_2) - \partial_{t_2}(\hat{q}(t_1,t_2)\hat{\phi}_L(t_1,t_2))|^2 \, dt_1 \, dt_2 \\
&:= I_1 + I_2.
\end{aligned}$$

For the first term, take derivatives to get

$$\begin{aligned}
I_1 &= \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\partial_{t_1}\hat{q}(t_1,t_2) - \partial_{t_1}\hat{q}(t_1,t_2)\hat{\phi}_L(t_1,t_2) - \hat{q}(t_1,t_2)\partial_{t_1}\hat{\phi}_L(t_1,t_2)|^2 \, dt_1 \, dt_2 \\
&\lesssim \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\partial_{t_1}\hat{q}(t_1,t_2) - \partial_{t_1}\hat{q}(t_1,t_2)\hat{\phi}_L(t_1,t_2)|^2 \, dt_1 \, dt_2 \\
&+ \frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\hat{q}(t_1,t_2)\partial_{t_1}\hat{\phi}_L(t_1,t_2)|^2 \, dt_1 \, dt_2 \\
&\lesssim \|\omega_1 q(\omega_1,\omega_2) - (\omega_1 q) * \phi_L(\omega_1,\omega_2)\|_2^2 + \|\hat{q}(t_1,t_2)\partial_{t_1}\hat{\phi}_L(t_1,t_2)\|_2^2.
\end{aligned}$$

We use Lemma 2 to get

$$4\|\omega_1 q(\omega_1,\omega_2) - (\omega_1 q) * \phi_L(\omega_1,\omega_2)\|_2^2 \lesssim \|\omega_1 q(\omega_1,\omega_2)\|_2^2 L^4 + L^{4\wedge(2\alpha-2)}.$$

By how we defined ϕ_L, we also have $\hat{\phi}_L(t_1,t_2) = e^{-L^2(t_1^2+t_2^2)/2}$. Take the derivative with respect to t_1:

$$\partial_{t_1}\hat{\phi}(t_1,t_2) = -L^2 t_1 e^{-L^2(t_1^2+t_2^2)/2}$$

and get

$$\|\hat{q}(t_1,t_2)\partial_{t_1}\hat{\phi}_L(t_1,t_2)\|_2^2 \le L^3\|q\|_2^2.$$

The result is

$$I_1 \lesssim \|\omega_1 q(\omega_1,\omega_2)\|_2^2 L^4 + L^{4\wedge(2\alpha-2)} + L^3\|q\|_2^2.$$

An identical argument is used to get

$$I_2 \lesssim \|\omega_2 q(\omega_1, \omega_2)\|_2^2 L^4 + L^{4\wedge(2\alpha-2)} + L^3\|q\|_2^2.$$

Now combine I_1 and I_2. The previous work implies

$$\|\omega_1 q(\omega_1, \omega_2)\|_2^2 + \|\omega_2 q(\omega_1, \omega_2)\|_2^2 = \|rq(r, \theta)\|_2^2$$

and

$$\|r(q - q * \phi_L)(r, \theta)\|_2^2 \lesssim \|rq(r, \theta)\|_2^2 L^4 + L^{4\wedge(2\alpha-2)} + L^3\|q\|_2^2.$$

□

Along with these two lemmas, we will also need the following lemma.

Lemma 4. Suppose ε is a mean zero Gaussian white noise supported on $[-1/2, 1/2]$ with variance σ^2. For all $p > 0$ and $\omega \in \mathbb{R}$,

$$\mathbb{E}\left[|\hat{\varepsilon}(\omega)|^p\right] \lesssim_p \sigma^p.$$

Proof. We rewrite $\hat{\varepsilon}$ as an integral with respect to a Brownian motion:

$$\hat{\varepsilon}(\omega) = \int_{-1/2}^{1/2} e^{-i\omega x} dB_x.$$

Let

$$g_1(\omega) = \mathrm{Re}(\hat{\varepsilon}(\omega)) = \int_{-1/2}^{1/2} \cos(\omega x) dB_x,$$
$$g_2(\omega) = \mathrm{Im}(\hat{\varepsilon}(\omega)) = \int_{-1/2}^{1/2} \sin(\omega x) dB_x.$$

For fixed ω, the random vector $\begin{pmatrix} g_1(\omega) \\ g_2(\omega) \end{pmatrix} \sim N(0, \Sigma(\omega))$ where the covariance matrix is given by

$$\Sigma(\omega) = \sigma^2 \begin{pmatrix} 1 + \frac{\sin\omega\cos\omega}{\omega} & 0 \\ 0 & 1 - \frac{\sin\omega\cos\omega}{\omega} \end{pmatrix}.$$

We will now prove a bound for $g_1(\omega)$. An identical bound applies for $g_2(\omega)$. Since $g_1(\omega)$ is normal,

$$\begin{aligned}
\mathbb{E}[|g_1(\omega)|^p] &= \sigma^p \frac{\mathrm{Var}^p(g_1(\omega))}{\sigma^{2p}} \frac{2^{p/2}\Gamma\left(\frac{p+1}{2}\right)}{\sqrt{\pi}} \\
&\leq \sigma^p \frac{\sigma^{2p}\left(1 - \frac{\sin\omega\cos\omega}{\omega}\right)^p}{\sigma^{2p}} \frac{2^{p/2}\Gamma\left(\frac{p+1}{2}\right)}{\sqrt{\pi}} \\
&\lesssim \sigma^p.
\end{aligned}$$

Now we have

$$\begin{aligned}
\mathbb{E}[|\hat{\varepsilon}(\omega)|^p] &= \mathbb{E}\left[\left(|\hat{\varepsilon}(\omega)|^2\right)^{p/2}\right] \\
&= \mathbb{E}\left[(g_1^2(\omega) + g_2^2(\omega))^{p/2}\right] \\
&\lesssim_p (\mathbb{E}\left[|g_1^2(\omega)|^{\frac{p}{2}}\right] + \mathbb{E}\left[|g_2^2(\omega)|^{\frac{p}{2}}\right]) \\
&\lesssim_p \sigma^p.
\end{aligned}$$

□

Lemma 5. Assume that the assumptions of Model 2 hold, $Bf \in C^3(\mathbb{R}^2)$, $(\cdot)\widehat{Bf}(\cdot)$ decays like $|\cdot|^\kappa$ for some $\kappa \geq 2$,

$$r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)| \in L^2(\mathbb{R}^2, dr \times d\theta),$$

and

$$r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)| \in L^2(\mathbb{R}^2, dr \times d\theta).$$

We have the bound

$$\|\tilde{g}_\sigma\|^2_{L^2(\Omega)} \lesssim_{\Omega,\tau} \frac{\sigma^2}{M}\|f\|_2^4 + \frac{\sigma^4}{M}\|f\|_2^2 + \frac{\sigma^6}{M}.$$

Proof. We use triangle inequality to get

$$\begin{aligned}
\|\tilde{g}_\sigma\|^2_{L^2(\Omega)} \lesssim & \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{f}_j(\omega_2) \hat{\varepsilon}_j(\omega_2 - \omega_1) \right\|^2_{L^2(\Omega)} + \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j^*(\omega_2) \hat{f}_j(\omega_2 - \omega_1) \hat{\varepsilon}_j(\omega_1) \right\|^2_{L^2(\Omega)} \\
& + \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{f}_j(\omega_2 - \omega_1) \hat{\varepsilon}_j^*(\omega_2) \right\|^2_{L^2(\Omega)} + \left\| \frac{1}{M} \sum_{j=1}^{M} B\varepsilon_j(\omega_1, \omega_2) \right\|^2_{L^2(\Omega)} \\
& + \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{\varepsilon}_j^*(\omega_2) \hat{\varepsilon}_j(\omega_2 - \omega_1) - h(\omega_1) \tilde{\mu}(\omega_1) \right\|^2_{L^2(\Omega)} \\
& + \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j^*(\omega_2) \hat{\varepsilon}_j(\omega_1) \hat{\varepsilon}_j(\omega_2 - \omega_1) - h(\omega_2) \tilde{\mu}^*(\omega_2) \right\|^2_{L^2(\Omega)} \\
& + \left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_2 - \omega_1) \hat{\varepsilon}_j(\omega_1) \hat{\varepsilon}_j^*(\omega_2) - h(\omega_2 - \omega_1) \tilde{\mu}(\omega_2 - \omega_1) \right\|^2_{L^2(\Omega)}.
\end{aligned}$$

The argument for bounding the expectation of the first three terms is similar, so we only provide the proof for the first term $\left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{f}_j^*(\omega_2) \hat{\varepsilon}_j(\omega_2 - \omega_1) \right\|^2_{L^2(\Omega)}$. We have

$$\begin{aligned}
& \mathbb{E}\left[\left\| \frac{1}{M} \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{f}_j^*(\omega_2) \hat{\varepsilon}_j(\omega_2 - \omega_1) \right\|^2_{L^2(\Omega)} \right] \\
&= \int_\Omega \mathbb{E}\left[\frac{1}{M^2} \left| \sum_{j=1}^{M} \hat{f}_j(\omega_1) \hat{f}_j(\omega_2) \hat{\varepsilon}_j(\omega_2 - \omega_1) \right|^2 \right] d\omega_1 \, d\omega_2 \\
&= \frac{\sigma^2}{M^2} \int_\Omega \sum_{j=1}^{M} |\hat{f}_j(\omega_1) \hat{f}_j(\omega_2)|^2 \, d\omega_1 \, d\omega_2 \\
&= \frac{\sigma^2}{M^2} \sum_{j=1}^{M} \int_\Omega |\hat{f}_j(\omega_1) \hat{f}_j(\omega_2)|^2 \, d\omega_1 \, d\omega_2 \\
&= \frac{\sigma^2}{M^2} \sum_{j=1}^{M} \int_\Omega |\hat{f}_j(\omega_1)|^2 \, d\omega_1 \int_\Omega |\hat{f}_j(\omega_2)|^2 \, d\omega_2 \\
&= \frac{\sigma^2}{M} \|\hat{f}\|^4_{L^2(\Omega)} \\
&\leq \frac{\sigma^2}{M} \|f\|\|^4_{L^2(\mathbb{R})},
\end{aligned}$$

where the last line follows from Phlancherel Theorem. A similar argument proves that each of the first three terms on the right are bounded on the order of $\frac{\sigma^2}{M}\|f\|_2^4$.

For the fourth term, since $\mathbb{E}[B\epsilon_j] = 0$, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} B\varepsilon_j(\omega_1,\omega_2)\right\|^2_{L^2(\Omega)}\right] &= \mathbb{E}\left[\int_\Omega \left|\frac{1}{M}\sum_{j=1}^{M} B\varepsilon_j(\omega_1,\omega_2)\right|^2 d\omega_1\, d\omega_2\right] \\
&= \int_\Omega \mathbb{E}\left[\left|\frac{1}{M}\sum_{j=1}^{M} B\varepsilon_j(\omega_1,\omega_2)\right|^2\right] d\omega_1\, d\omega_2 \\
&= \int_\Omega \mathrm{Var}\left[\frac{1}{M}\sum_{j=1}^{M} B\varepsilon_j(\omega_1,\omega_2)\right] d\omega_1\, d\omega_2 \\
&= \frac{1}{M}\int_\Omega \mathrm{Var}(B\varepsilon_j)\, d\omega_1\, d\omega_2.
\end{aligned}$$

We now bound $\mathrm{Var}(B\varepsilon_j)$. First, since the expectation is zero, by Holder's inequality and the Lemma 4 to get

$$\begin{aligned}
\mathrm{Var}(B\varepsilon_j) &= \mathbb{E}[|B\varepsilon_j|^2] \\
&= \mathbb{E}\left[|\hat{\varepsilon}(\omega_1)|^2|\hat{\varepsilon}(\omega_2)|^2|\hat{\varepsilon}(\omega_2-\omega_1)|^2\right] \\
&\le \mathbb{E}\left[|\hat{\varepsilon}(\omega_1)|^6\right]^{1/3}\mathbb{E}\left[|\hat{\varepsilon}(\omega_2)|^6\right]^{1/3}\mathbb{E}\left[|\hat{\varepsilon}(\omega_2-\omega_1)|^6\right]^{1/3} \\
&\lesssim \sigma^6.
\end{aligned}$$

Thus, we obtain

$$\mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} B\varepsilon_j(\omega_1,\omega_2)\right\|^2_{L^2(\Omega)}\right] \lesssim \frac{\sigma^6}{M}|\Omega|.$$

Now we bound the last three terms. We start with

$$A = \mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} \hat{f}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1) - h(\omega_1)\tilde{\mu}(\omega_1)\right\|^2_{L^2(\Omega)}\right].$$

Consider the random variable

$$A_j = \hat{f}_j(\omega_1)\hat{\varepsilon}_j^*(\omega_2)\hat{\varepsilon}_j(\omega_2-\omega_1).$$

Then

$$
\begin{aligned}
A &= \mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} A_j - h(\omega_1)\tilde{\mu}(\omega_1)\right\|^2_{L^2(\Omega)}\right] \\
&= \mathbb{E}\left[\left\|\left(\frac{1}{M}\sum_{j=1}^{M} A_j - h(\omega_1)\mu(\omega_1)\right) + h(\omega_1)\mu(\omega_1) - h(\omega_1)\tilde{\mu}(\omega_1)\right\|^2_{L^2(\Omega)}\right] \\
&\lesssim \mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} A_j - h(\omega_1)\mu(\omega_1)\right\|^2_{L^2(\Omega)}\right] + \mathbb{E}\left[\|h(\omega_1)\mu(\omega_1) - h(\omega_1)\tilde{\mu}(\omega_1)\|^2_{L^2(\Omega)}\right].
\end{aligned}
$$

For the first term at the end of the inequality, we see that $h(\omega_1)\mu(\omega_1)$ is the mean of A_j for fixed (ω_1, ω_2). Thus,

$$
\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{M}\sum_{j=1}^{M} A_j - h(\omega_1)\mu(\omega_1)\right\|^2_{L^2(\Omega)}\right] &= \int_\Omega \mathbb{E}\left[\left|\frac{1}{M}\sum_{j=1}^{M} A_j - h(\omega_1)\mu(\omega_1)\right|^2\right] d\omega_1\, d\omega_2 \\
&= \int_\Omega \frac{1}{M}\mathrm{Var}\left[A_j\right]\, d\omega_1\, d\omega_2
\end{aligned}
$$

We have

$$
\begin{aligned}
\mathrm{Var}(A_j) &= \mathbb{E}[|\hat{f}_j(\omega_1)\hat{\varepsilon}^*_j(\omega_2)\hat{\varepsilon}_j(\omega_2 - \omega_1)|^2] - h^2(\omega_1)\mu^2(\omega_1) \\
&\leq \mathbb{E}[|\hat{f}_j(\omega_1)\hat{\varepsilon}^*_j(\omega_2)\hat{\varepsilon}_j(\omega_2 - \omega_1)|^2] \\
&\lesssim \sigma^4 \mathbb{E}_{t,\tau}[|\hat{f}_j(\omega_1)|^2].
\end{aligned}
$$

Now substitute this back into the integral to get

$$
\int_\Omega \frac{1}{M}\mathrm{Var}\left[A_j\right]\, d\omega_1\, d\omega_2 \lesssim \frac{\sigma^4}{M}\int_\Omega \mathbb{E}_{t,\tau}[|\hat{f}_j(\omega_1)|^2]\, d\omega_1\, d\omega_2 = \frac{\sigma^4}{M}\int_\Omega \mathbb{E}_\tau\left[|\hat{f}_j(\omega_1)|^2\right]\, d\omega_1\, d\omega_2
$$

by translation invariance of the power spectrum. Now we have

$$
\begin{aligned}
\frac{\sigma^4}{M}\int_\Omega \mathbb{E}_\tau\left[|\hat{f}_j(\omega_1)|^2\right]\, d\omega_1\, d\omega_2 &= \frac{\sigma^4}{M}\mathbb{E}_\tau\left[\int_\Omega |\hat{f}_j(\omega_1)|^2\, d\omega_1\, d\omega_2\right] \\
&= \frac{\sigma^4}{M}\mathbb{E}_\tau\left[\frac{1}{1-\tau_j}\int\int |\hat{f}(\tilde{\omega}_1)|^2\, d\tilde{\omega}_1\, d\omega_2\right] \\
&\lesssim_{\tau,\Omega} \frac{\sigma^4}{M}\|f\|_2^2.
\end{aligned}
$$

For the second term,

$$\begin{aligned}\mathbb{E}\left[\|h(\omega_1)\mu(\omega_1) - h(\omega_1)\tilde{\mu}(\omega_1)\|^2_{L^2(\Omega)}\right] &= \int_\Omega h^2(\omega_1)\mathbb{E}\left[|\mu(\omega_1) - \tilde{\mu}(\omega_1)|^2\right]\, d\omega_1\, d\omega_2 \\ &\lesssim \sigma^4 \int_\Omega \mathbb{E}\left[|\mu(\omega_1) - \tilde{\mu}(\omega_1)|^2\right]\, d\omega_1\, d\omega_2 \\ &= \sigma^4 \int_\Omega \mathbb{E}\left[\left|\frac{1}{M}\sum_{j=1}^M \hat{y}_j(\omega_1) - \mu(\omega_1)\right|^2\right] d\omega_1\, d\omega_2\end{aligned}$$

Define $Z_j = \hat{y}_j(\omega) - \mu(\omega)$. Using similar steps to before, we get

$$\begin{aligned}\mathbb{E}\left[\|h(\omega_1)\mu(\omega_1) - h(\omega_1)\tilde{\mu}(\omega_1)\|^2_{L^2(\Omega)}\right] &\lesssim \sigma^4 \int_\Omega \left(\frac{1}{M}\sum_{j=1}^M Z_j\right)^2 d\omega_1\, d\omega_2 \\ &= \frac{\sigma^4}{M}\int_\Omega \mathrm{Var}(Z_j)\, d\omega_1\, d\omega_2 \\ &\lesssim_\Omega \frac{\sigma^4}{M}(\|f\|_2^2 + \sigma^2).\end{aligned}$$

This means that we put everything together to conclude

$$\|\tilde{g}_\sigma\|^2_{L^2(\Omega)} \lesssim_{\Omega,\tau} \frac{\sigma^2}{M}\|f\|_2^4 + \frac{\sigma^4}{M}\|f\|_2^2 + \frac{\sigma^6}{M}.$$

□

Theorem 6. Assume that the assumptions of Model 2 hold, $Bf \in C^3(\mathbb{R}^2)$, $(\cdot)\widehat{Bf}(\cdot)$ decays like $|\cdot|^\kappa$ for some $\kappa \geq 2$,

$$r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)| \in L^2(\mathbb{R}^2, dr \times d\theta),$$

and

$$r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)| \in L^2(\mathbb{R}^2, dr \times d\theta).$$

For the estimator $(\widetilde{Bf})(r,\theta)$ defined for Model 2,

$$\mathbb{E}\left[\|Bf - \widetilde{Bf}\|^2_{L^2(\Omega)}\right] \leq C_{f,\Omega}\left(\frac{\eta^2}{M} + L^4 + \frac{\sigma^2 \vee \sigma^6}{L^2 M}\right),$$

where $C_{f,\Omega}$ only depends on f and Ω.

Proof. First, we see that by an argument as in Lemma 1,

$$\begin{aligned}
\|Bf - \widetilde{Bf}\|_{L^2(\Omega)}^2 &\lesssim \left\|4g_\eta + r\partial_r g_\eta - 4(\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L - r\partial_r((\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L)\right\|_{L^2(\Omega)}^2 \\
&\lesssim \left\|g_\eta - \tilde{g}_\eta\right\|_{L^2(\Omega)}^2 + \left\|r\partial_r g_\eta - r\partial_r \tilde{g}_\eta\right\|_{L^2(\Omega)}^2 \\
&\quad + \left\|4\tilde{g}_\eta + r\partial_r \tilde{g}_\eta - 4(\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L - r\partial_r((\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L)\right\|_{L^2(\Omega)}^2 \\
&\lesssim \left\|g_\eta - \tilde{g}_\eta\right\|_{L^2(\Omega)}^2 + \left\|r\partial_r g_\eta - r\partial_r \tilde{g}_\eta\right\|_{L^2(\Omega)}^2 + \|\tilde{g}_\eta - \tilde{g}_\eta * \phi_L\|_{L^2(\Omega)}^2 \\
&\quad + \left\|r\partial_r \tilde{g}_\eta - r\partial_r(\tilde{g}_\eta * \phi_L)\right\|_{L^2(\Omega)}^2 + \|\tilde{g}_\sigma * \phi_L\|_{L^2(\Omega)}^2 + \|r\partial_r(\tilde{g}_\sigma * \phi_L)\|_{L^2(\Omega)}^2
\end{aligned}$$

By Theorem 1,

$$\begin{aligned}
&\mathbb{E}\left[\left\|g_\eta - \tilde{g}_\eta\right\|_{L^2(\Omega)}^2 + \left\|r\partial_r g_\eta - r\partial_r \tilde{g}_\eta\right\|_{L^2(\Omega)}^2\right] \\
&\lesssim \frac{\eta^2}{M}\left(\|(Bf)(r,\theta)\|_2^2 + 2\|r\partial_r(Bf)(r,\theta)\|_2^2 + \|r^2\partial_{rr}(Bf)(r,\theta)]^2\|_2^2\right) \\
&+ \frac{\eta^4}{M}\left(\left\|r^2 \max_{\alpha\in[r/2,2r]} |\partial_{\alpha\alpha}(Bf)(\alpha,\theta)|\right\|_2^2 + \left\|r^3 \max_{\gamma\in[r/2,2r]} |\partial_{\gamma\gamma\gamma}(Bf)(\gamma,\theta)|\right\|_2^2\right).
\end{aligned}$$

It suffices to bound the other four terms appropriately now. By Lemma 2,

$$\|\tilde{g}_\eta - \tilde{g}_\eta * \phi_L\|_{L^2(\Omega)}^2 \lesssim \|\tilde{g}_\eta\|_2^2 L^4 + L^{4\wedge(2\kappa-2)}.$$

For $\tilde{g}_\eta$, notice that

$$\begin{aligned}
\|Bf_j\|_2^2 &= (1-\tau_j)^6 \int_{\mathbb{R}^2} |(Bf)((1-\tau_j)\omega_1, (1-\tau_j)\omega_2)|^2 \, d\omega_1 d\omega_2 \\
&= (1-\tau_j)^4 \int_{\mathbb{R}^2} |(Bf)(\omega_1,\omega_2)|^2 \, d\omega_1 d\omega_2 \\
&= (1-\tau_j)^4 \|Bf\|_2^2 \\
&\leq 2^4 \|Bf\|_2^2.
\end{aligned}$$

Triangle inequality implies

$$\|\tilde{g}_\eta\|_2 = \left\|\frac{1}{M}\sum_{j=1}^M Bf_j\right\|_2 \leq \frac{1}{M}\sum_{j=1}^M \|Bf_j\|_2 \lesssim \|Bf\|_2$$

and

$$\|\tilde{g}_\eta - \tilde{g}_\eta * \phi_L\|_{L^2(\Omega)} \lesssim \|Bf\|_2^2 L^4 + L^{4\wedge(2\alpha-2)}.$$

Also, by Lemma 3,

$$\|r\partial_r(\tilde{g}_\eta - (\tilde{g}_\eta * \phi_L))\|^2_{L^2(\Omega)} \lesssim \|r\partial_r \tilde{g}_\eta\|_2^2 L^4 + L^{4\wedge(2\alpha-2)} + L^3\|\partial_r \tilde{g}_\eta\|_2^2.$$

Now consider the error terms. First start with $\|\tilde{g}_\sigma * \phi_L\|^2_{L^2(\Omega)}$. We have

$$\|\tilde{g}_\sigma * \phi_L\|^2_{L^2(\Omega)} \leq \|\phi_L\|^2_{L^1(\Omega)}\|\tilde{g}_\sigma\|^2_{L^2(\Omega)} \lesssim \|\tilde{g}_\sigma\|^2_{L^2(\Omega)}.$$

Now we consider $\|r\partial_r(\tilde{g}_\sigma * \phi_L)\|_{L^2(\Omega)}$. Let $R_\Omega = \max_\Omega |r|$. We have

$$\begin{aligned}\|r\partial_r(\tilde{g}_\sigma * \phi_L)\|^2_{L^2(\Omega)} &\leq R_\Omega^2\|\tilde{g}_\sigma * \partial_r\phi_L\|^2_{L^2(\Omega)} \\ &\leq R_\Omega^2\|\partial_r\phi_L\|_1^2\|\tilde{g}_\sigma\|^2_{L^2(\Omega)}.\end{aligned}$$

Since ϕ is a radial filter and we know the filter

$$\partial_r\phi_L(r) = (2\pi L^2)^{-1/2}\partial_r\left[e^{-r^2/(2L^2)}\right] = -(2\pi L^3)^{-1/2} r e^{-r^2/(2L^2)}.$$

It follows that

$$\|\partial_r\phi_L\|_1 = \frac{1}{L^3}\int_0^\infty r e^{-r^2/(2L^2)}\, dr = L^{-1}$$

and $\|\partial_r\phi_L\|_1^2 = L^{-2}$. We can also use our previous bound to get

$$\|r\partial_r(\tilde{g}_\sigma * \phi_L)\|^2_{L^2(\Omega)} \lesssim_{\tau,\Omega,f} L^{-2}\left(\frac{\sigma^2 \vee \sigma^6}{M}\right).$$

This finishes the proof of the theorem since each term is dependent on L, M, and η^2 now. □

Note that we can choose $L = \frac{\sigma}{M^{1/6}}$ to get a bound of $O\left(\frac{\sigma^6}{M}\right)$ when $\sigma \geq 1$. We let $L^4 = \frac{\sigma^6}{L^2 M}$, and this gives a convergence rate of $O(\frac{\eta^2}{M} + \frac{\sigma^4}{M^{2/3}})$ on the squared error.

2.7 Numerical Implementation of Bispectrum Recovery

Now that we have theoretical results to recover the bispectrum, we will use these results for bispectrum recovery. After showing success for signal recovery, we will use our recovery results for signal inversion with the phase synchronization algorithm from [7], which will require a recovered power spectrum and recovered bispectrum, which will motivate our current approach of doing power spectrum recovery first.

We can discuss computing (2.12) numerically now. We cannot compute $(I - L_{C_0})^{-1}$, but we can still solve for the bispectrum by means of optimization. Similar to how we constructed the estimator in the finite sample case, we will consider the infinite sample case first and derive an optimization procedure. Based on this procedure, we will design another optimization procedure for the finite sample case that is a good estimator when M is large.

In the infinite sample limit, we have access to the term

$$d(r, \theta) = 4g_\eta(r, \theta) + r\partial_r g_\eta(r, \theta) \tag{2.13}$$

and Proposition 4 implies that we can recover d by solving the convex optimization problem

$$g = \operatorname{argmin}_{\tilde{g}} \|(I - L_{C_0})\tilde{g} - C_1 L_{C_2} d\|_2^2, \tag{2.14}$$

where the constants C_0, C_1, C_2 depend on η as given in (2.5). Note that for fixed η, this problem is convex. Since the variance of the dilations is not necessarily known, η is possibly an unknown parameter and we must actually minimize

$$\mathfrak{Q}(\tilde{g}, \tilde{\eta}) = \left\|(I - L_{C_0(\tilde{\eta})})\tilde{g} - C_1(\tilde{\eta}) L_{C_2(\tilde{\eta})} d\right\|_2^2 .$$

In [26], the authors find η via a nonconvex optimization problem while recovering the power spectrum. We could mimic a similar process for bispectrum recovery, but it is likely this process would fail. The reason for this is that the optimization problem for bispectrum recovery has a larger number of variables, so memory limited methods are necessary. In addition, we also need a recovered power spectrum for our bispectrum inversion algorithm anyway, so we would still need to perform power spectrum recovery. We will choose to learn η and recover the power spectrum using a modification of the algorithm from [26] and describe the steps below.

For recovering the power spectrum, our data term is given by

$$p_{\text{data}}(\omega) = 3p_\eta(\omega) + \omega p'_\eta(\omega) , \tag{2.15}$$

and we can consider minimizer

$$p = \operatorname{argmin}_{\tilde{p}} \frac{\|(I - S_{C_0})\tilde{p} - C_1 S_{C_2} p_{\text{data}}\|_2^2}{\eta^2} , \tag{2.16}$$

Since η may be unknown, we use the loss function:

$$\mathcal{L}(\tilde{p}, \tilde{\eta}) = \frac{\| \left(I - S_{C_0(\tilde{\eta})}\right) \tilde{p} - C_1(\tilde{\eta}) S_{C_2(\tilde{\eta})} p_{\text{data}} \|_2^2}{\eta^2} \,. \tag{2.17}$$

Theorem 7. Define the operator $A = I - S_{C_0}$. For the loss function in (2.17), we have

$$\begin{aligned}
\nabla_{\tilde{p}} \mathcal{L}(\tilde{p}, \tilde{\eta}) &= \frac{2A^*(A\tilde{p} - C_1 S_{C_2} p_{\text{data}})}{\eta^2} \\
\nabla_{\tilde{\eta}} \mathcal{L}(\tilde{p}, \tilde{\eta}) &= \frac{1}{\eta^2} \int 2(A\tilde{p}(\omega) - C_1 S_{C_2} p_{\text{data}}(\omega)) \frac{\partial}{\partial \tilde{\eta}} (A\tilde{p}(\omega) - C_1 S_{C_2} p_{\text{data}}(\omega)) \, d\omega \\
&\quad - \frac{2}{\eta^3} \|A\tilde{p} - C_1 S_{C_2} p_{\text{data}}\|_2^2.
\end{aligned}$$

Proof. The following computation is almost identical to [26], but we provide the details for completeness. We start with $\nabla_{\tilde{p}} \mathcal{L}(\tilde{p}, \tilde{\eta})$ and fix $\tilde{\eta}$. We will ignore the $\tilde{\eta}$ dependence for this part of the computation. The Frechet derivative is

$$\mathcal{L}(\tilde{p}) = \frac{\|A\tilde{p} - C_1 S_{C_2} p_{\text{data}}\|_2^2}{\eta^2} = N(A\tilde{p}) \,,$$

where $Nf = \|f - C_1 S_{C_2} p_{\text{data}}\|_2^2$. Let h be a test function. Then we have

$$(D\mathcal{L})(\tilde{p})h = (DN)(A\tilde{p}) \circ D(A\tilde{p})h = (DN)(A\tilde{p}) \circ Ah$$

since A is a linear operator. It follows that

$$\frac{|N(f+h) - Nf - \frac{2}{\eta^2}\langle f - C_1 S_{C_2} p_{\text{data}}, h\rangle|}{\|h\|_2} = \frac{1}{\eta^2} \frac{\|h\|_2^2}{\|h\|_2} \to 0$$

as $\|h\|_2 \to 0$, and $(DN)(f)h = \frac{2}{\eta^2}\langle f - C_1 S_{C_2} p_{\text{data}}, h\rangle$. Thus

$$\begin{aligned}
(D\mathcal{L})(\tilde{p})h &= \frac{2}{\eta^2}\langle A\tilde{p} - C_1 S_{C_2} p_{\text{data}}, Ah\rangle \\
&= \langle \frac{2}{\eta^2} A^*(A\tilde{p} - C_1 S_{C_2} p_{\text{data}}), h\rangle.
\end{aligned}$$

It now follows that

$$\nabla \mathcal{L}(\tilde{p}) = \frac{2}{\eta^2} A^*(A\tilde{p} - C_1 S_{C_2} p_{\text{data}}) \,. \tag{2.18}$$

Additionally, using the work above, we have

$$\begin{aligned}\nabla_{\tilde{\eta}}\mathcal{L}(\tilde{p},\tilde{\eta}) &= \frac{\eta^2\nabla_\eta(\|A\tilde{p}-C_1S_{C_2}p_{\text{data}}\|_2^2)-2\eta\|A\tilde{p}-C_1S_{C_2}p_{\text{data}}\|_2^2}{\eta^4}\\ &= \frac{1}{\eta^2}\int 2(A\tilde{p}(\omega)-C_1S_{C_2}p_{\text{data}}(\omega))\frac{\partial}{\partial\tilde{\eta}}(A\tilde{p}(\omega)-C_1S_{C_2}p_{\text{data}}(\omega))\,d\omega\\ &\quad-\frac{2}{\eta^3}\|A\tilde{p}-C_1S_{C_2}p_{\text{data}}\|_2^2.\end{aligned}$$

□

For this implementation, we need to calculate A^*, which we do analytically below. We have

$$\begin{aligned}\langle Ag,h\rangle &= \int Ag(\omega)h(\omega)\,d\omega\\ &= \int (g(\omega)-C_0^3g(C_0\omega))h(\omega)\,d\omega\\ &= \int g(\tilde{\omega})\left(h(\tilde{\omega})-C_0^2h\left(\frac{\tilde{\omega}}{C_0}\right)\right)\,d\tilde{\omega}\\ &= \langle g,A^*h\rangle.\end{aligned}$$

Thus,

$$A^*h(\omega) = h(\omega)-C_0^2h\left(\frac{\omega}{C_0}\right). \tag{2.19}$$

For implementations, one only has access to the estimate of p_η, so numerical implementations will use the following estimate for the data term:

$$\begin{aligned}\widetilde{p}_{\text{data}}(\omega) &:= 3(\widetilde{p}_\eta+\widetilde{p}_\sigma)*\phi_L(\omega)+\omega\left[(\widetilde{p}_\eta+\widetilde{p}_\sigma)*\phi_L'\right](\omega)\\ \widetilde{\mathcal{L}}(\tilde{p},\tilde{\eta}) &:= \left\|\left(I-S_{C_0(\tilde{\eta})}\right)\tilde{p}-C_1(\tilde{\eta})S_{C_2(\tilde{\eta})}\widetilde{p}_{\text{data}}\right\|_2^2,\end{aligned}$$

and recovered power spectrum is computed by minimizing $\widetilde{\mathcal{L}}$.

After recovering the power spectrum and an estimate for η, the problem for recovering the bispectrum is

$$g = \text{argmin}_{\tilde{g}}\|(I-L_{C_0(\eta)})\tilde{g}-C_1(\eta)L_{C_2(\eta)}d\|_2^2. \tag{2.20}$$

for a fixed estimate $\tilde{\eta}$, which is a convex optimization problem. We let $Y = I - L_{C_0}$. By a proof identical to above, we get

$$\nabla_{\tilde{g}}\mathfrak{L}(\tilde{g}) = 2Y^*(Y\tilde{g}-C_1L_{C_2}d)$$

Additionally, we have

$$Y^* h(\omega_1, \omega_2) = h(\omega_1, \omega_2) - C_0^2 h(C_0^{-1} w_1, C_0^{-1} w_2)). \tag{2.21}$$

Like before, for numerical applications, we only have access to a finite number of samples, so will have to have to modify our data term and loss function based on the estimator provided in Model 2:

$$\begin{aligned}
\tilde{d}(r,\theta) &= 4(\tilde{g}_\eta + \tilde{g}_\sigma) * \phi_L(r,\theta) + r[(\tilde{g}_\eta + \tilde{g}_\sigma) * \partial_r \phi_L](r,\theta) \\
\tilde{\mathfrak{L}}(\tilde{g}) &= \left\| (I - L_{C_0})\tilde{g} - C_1 L_{C_2} \tilde{d} \right\|_2^2 .
\end{aligned}$$

With this surrogate loss and data term, we can conduct numerical experiments in the next section.

2.8 Numerical Experiments for Bispectrum Recovery

We will now test our bispectrum recovery algorithm on the following signals:

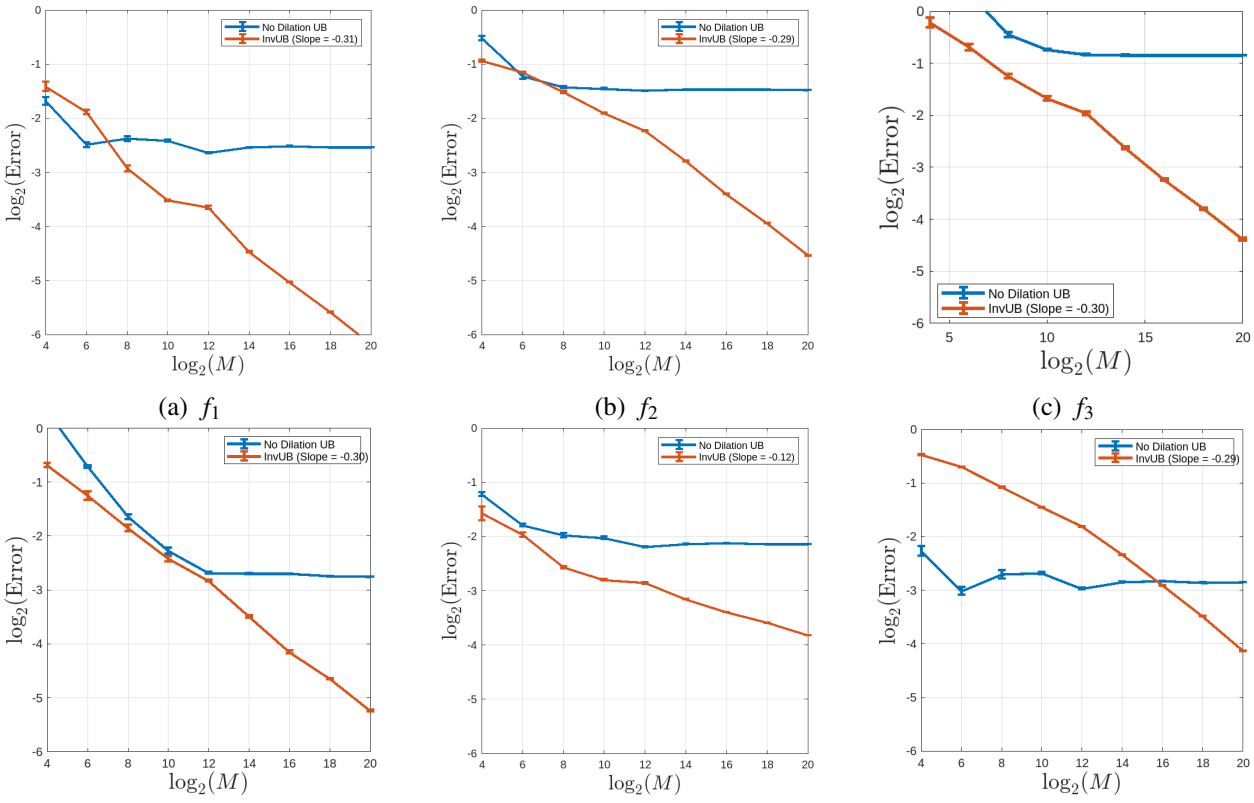
$$\begin{aligned}
f_1(x) &= A_1 e^{-5x^2} \cos(4x) \\
f_2(x) &= A_2 e^{-5x^2} \cos(8x) \\
f_3(x) &= A_3 e^{-5x^2} \cos(12x) \\
\hat{f_4}(x) &= A_4 1_{[-1/8,1/8]}(x) \\
f_5(x) &= A_5 \mathrm{sinc}(4x) \\
f_6(x) &= A_6 \begin{cases} 2 - 2|x|, & |x| < 1 \\ 0, & \text{otherwise} \end{cases} .
\end{aligned}$$

Like in [27], we define our hidden signals on $[-N/4, N/4]$, and the noisy signals are defined on $[-N/2, N/2]$ with $N = 2^5$. In frequency, the signals were sampled on the interval $[-2^\ell \pi, 2^\ell \pi]$ with $\ell = 4$ and sampling rate of π/N. The constants A_i with $i = 1, \ldots, 6$ were chosen so that the SNR for each f_i was set to be σ^{-2}, where $\text{SNR} = \left(\int_{-N/2}^{N/2} |f(x)|^2 \, dx \right) / \sigma^2$. All signals, other than f_4, were generated in space. We chose to generate f_4 in frequency. We chose to generate f_4 in frequency because of aliasing.

Regarding our signals, we chose our signals so that they could test the robustness of our proposed method. The signals f_1 to f_3 are smooth with fast decay, which do not fit the assumptions to employ

our proposed estimators. Nonetheless, f_1 to f_3 still perform well, which is most likely because of their exponential decay rate. However, we note that as the peak of the power spectrum is farther from the origin, our problem becomes much harder problem because the dilations cause larger perturbations. This is shown in Figure 2.4.

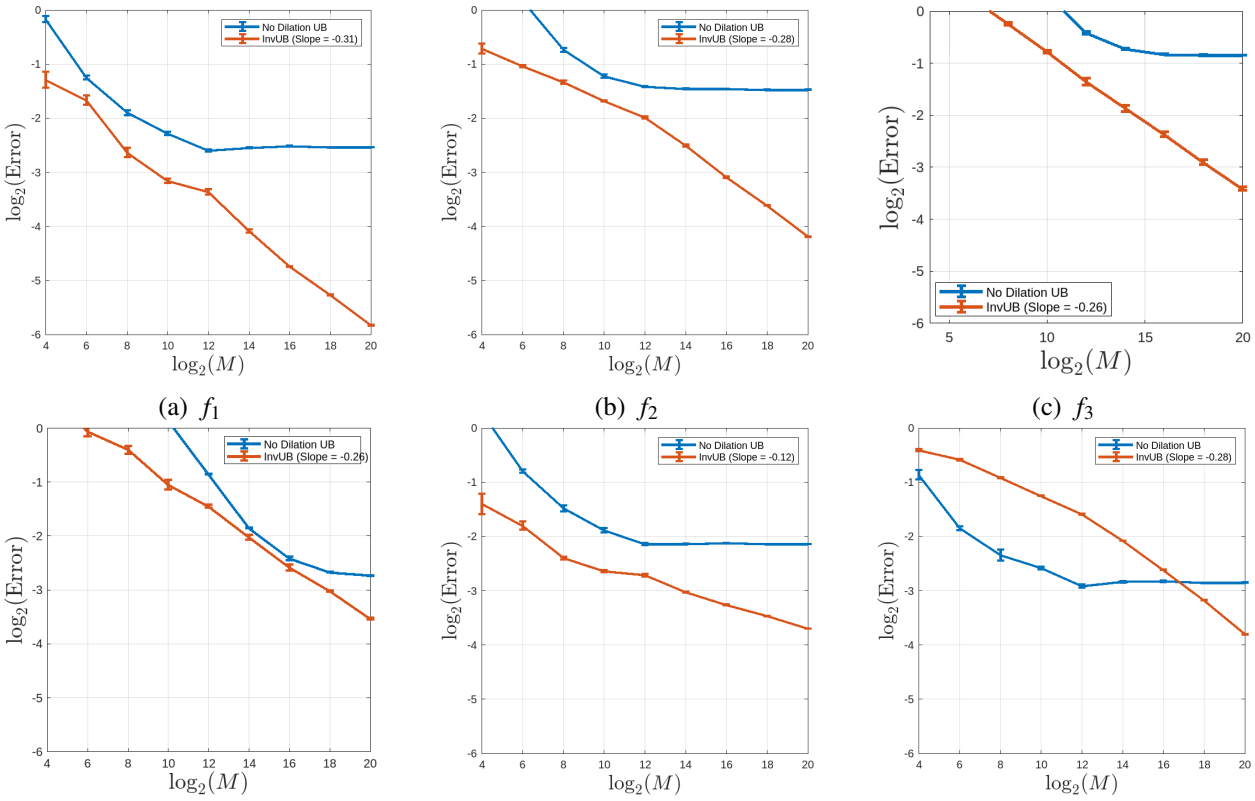
We start with the case of oracle η (i.e. η is known) and consider two cases: $\sigma = 0.5$ and $\sigma = 1.0$. Error plots are shown in Figures 2.4 and 2.5. All oracle experiments were run with $\eta = 12^{-1/2}$ and with a gaussian width of $L = 5(\frac{\sigma}{M})^{1/6}$ for bispectrum recovery and width $L = 10(\frac{\sigma^4}{M})^{1/6}$ for power spectrum recovery.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.4 Relative error decay with standard error bars for Bispectrum Recovery using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 0.5$.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.5 Relative error decay with standard error bars for Bispectrum Recovery using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ and $\sigma = 1.0$.

For all the signals, in the case where one only does empirical noise centering on the average bispectra, which is marked in blue, there is a diminishing return in performance for large enough M. This is most likely because the blue estimator has a dilation bias, and a large sample size will not be able to overcome this. It will only be able to overcome the additive noise bias, so eventually the blue line will plateau. In comparison, our inversion unbiasing procedure,which is marked in red, demonstrates a continual drop in error for both choices of σ, except for f_5. Additionally, note that the red decay line does not plateau, but is linear on the log-log plot, supporting the claim of Theorem 5 that $\widetilde{Bf}$ is truly an unbiased estimator. The poor performance of f_5 stems from the fact that it does not obey the assumptions of Model 2 since sinc is not a compactly supported function. The same is true for each of the Gabor functions, but their decay is exponential rather than polynomial like the sinc.

Bispectrum recovery examples for $\sigma = 0.5$ are given in Figures 2.6 and 2.7. Note that in all

the examples, once can "undilate" the additive noise unbiased bispectrum average. The results are similar for $\sigma = 1.0$, but they are not provided in this thesis.

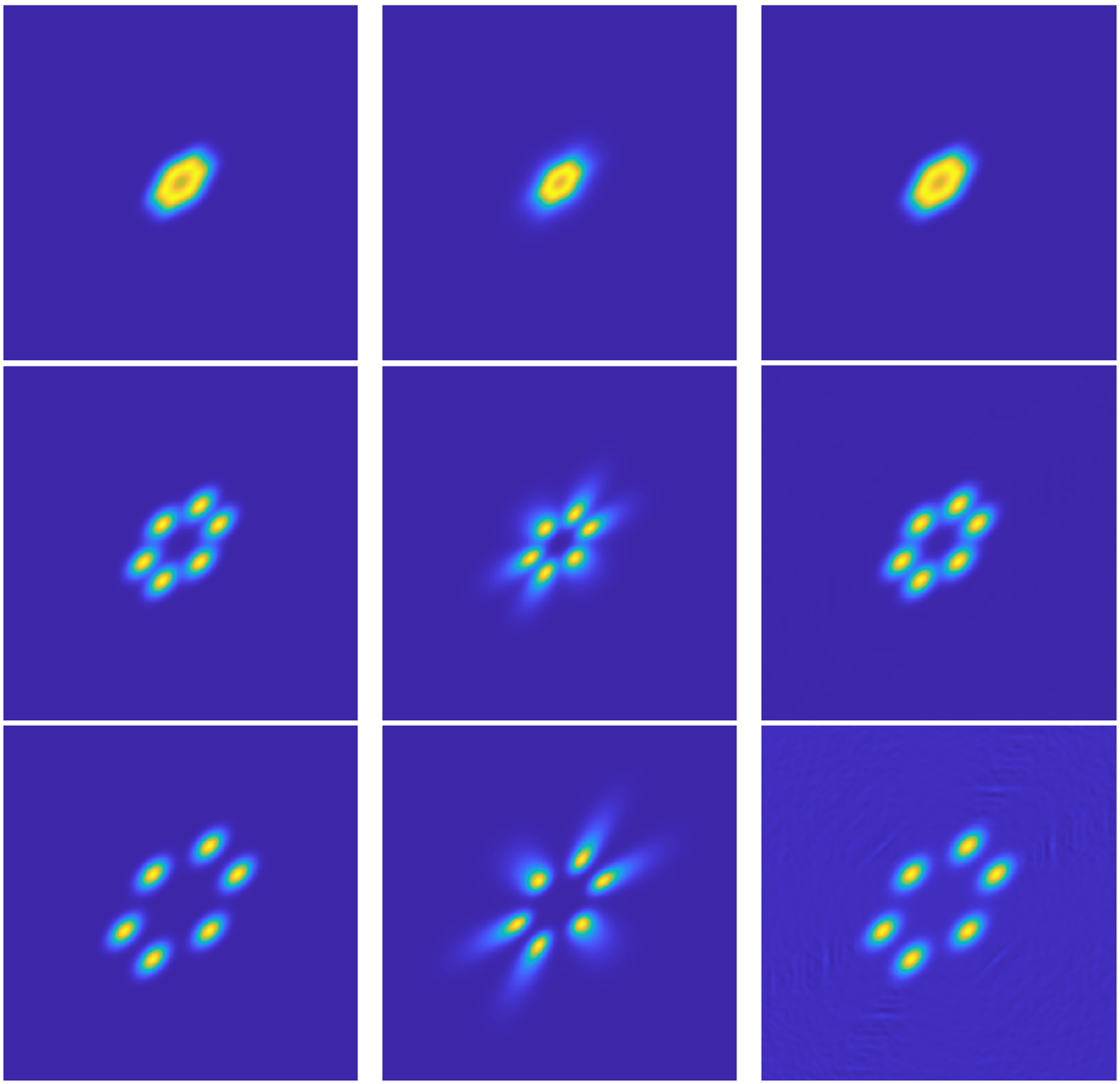


Figure 2.6 Example plots of recovery for f_1, f_2, and f_3 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 0.5$. Left: ground truth. Middle: Only Additive Noise Unbiasing. Our Unbiasing Procedure.

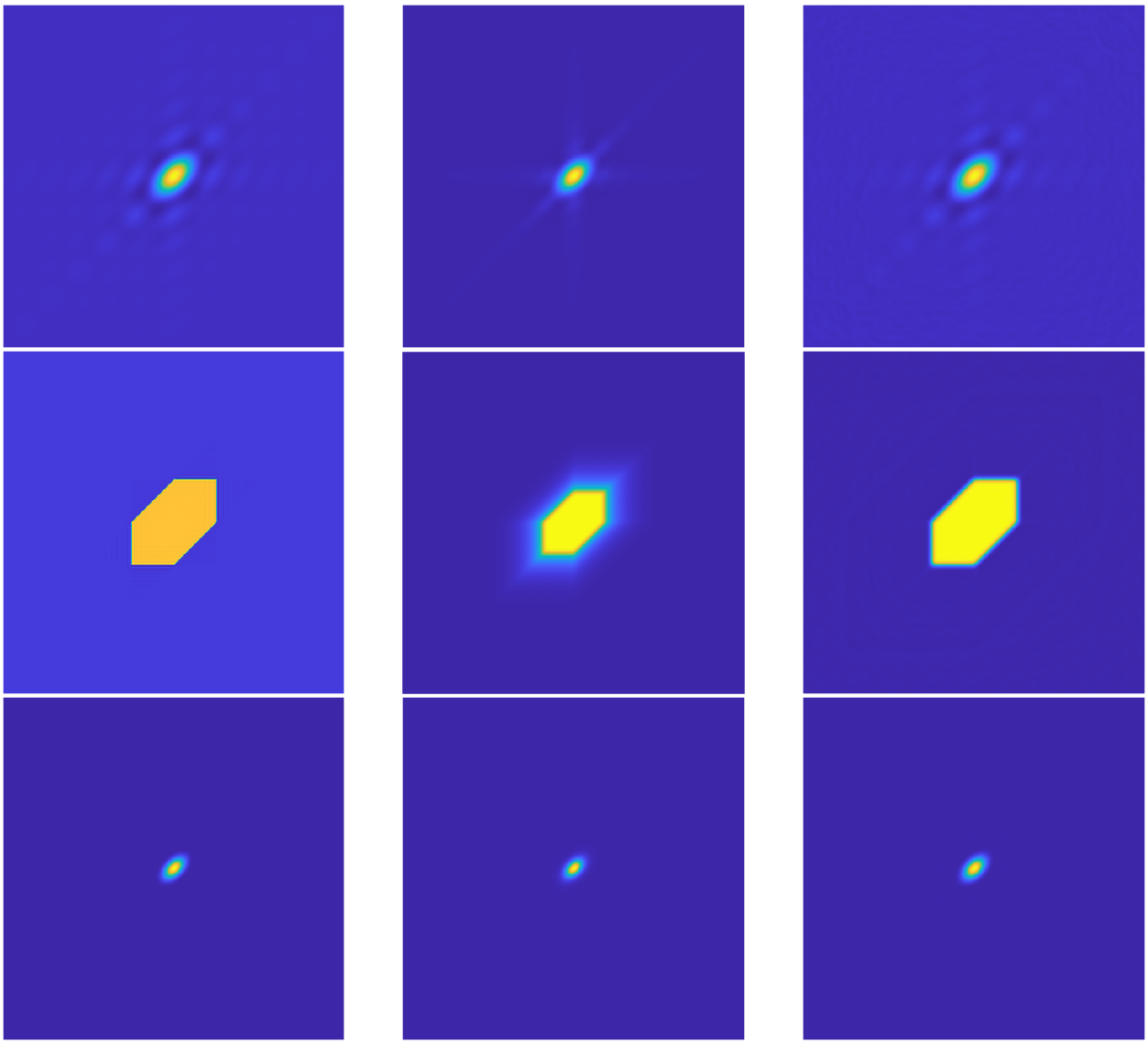
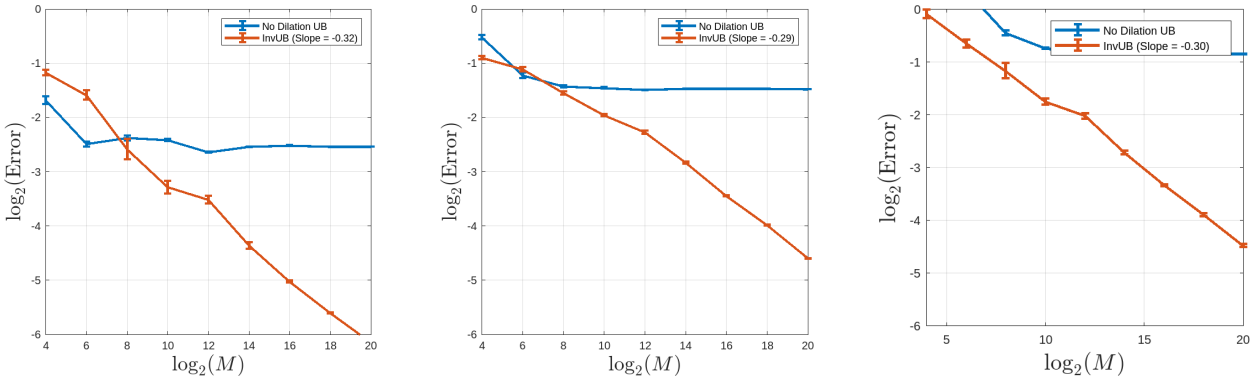


Figure 2.7 Example plots of recovery for f_4, f_5, and f_6 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 0.5$. Left: ground truth. Middle: Only Additive Noise Unbiasing. Our Unbiasing Procedure.

We now consider the case where η is unknown and estimated. Again, we have two cases: $\sigma = 0.5$, and $\sigma = 1.0$. For estimating σ, since all the signals decay away from the origin, the values of the power spectrum average should be near zero, sans noise. Thus, we find the variance of the function values on the edge of the signal. All experiments will be run with $\eta = 12^{-1/2}$. Since we now need to estimate the power spectrum, we let the width for the smoothing parameter for power spectrum recovery be $5(\frac{\sigma^4}{M})^{1/6}$ and use the smoothing parameter $5(\frac{\sigma}{M})^{1/6}$ for the bispectrum

recovery. Note that the smoothing parameter for the power spectrum is smaller than the oracle case. We found that $10(\frac{\sigma^4}{M})^{1/6}$ and $5(\frac{\sigma^4}{M})^{1/6}$ yielded similar results, so we decided to not change the parameter.

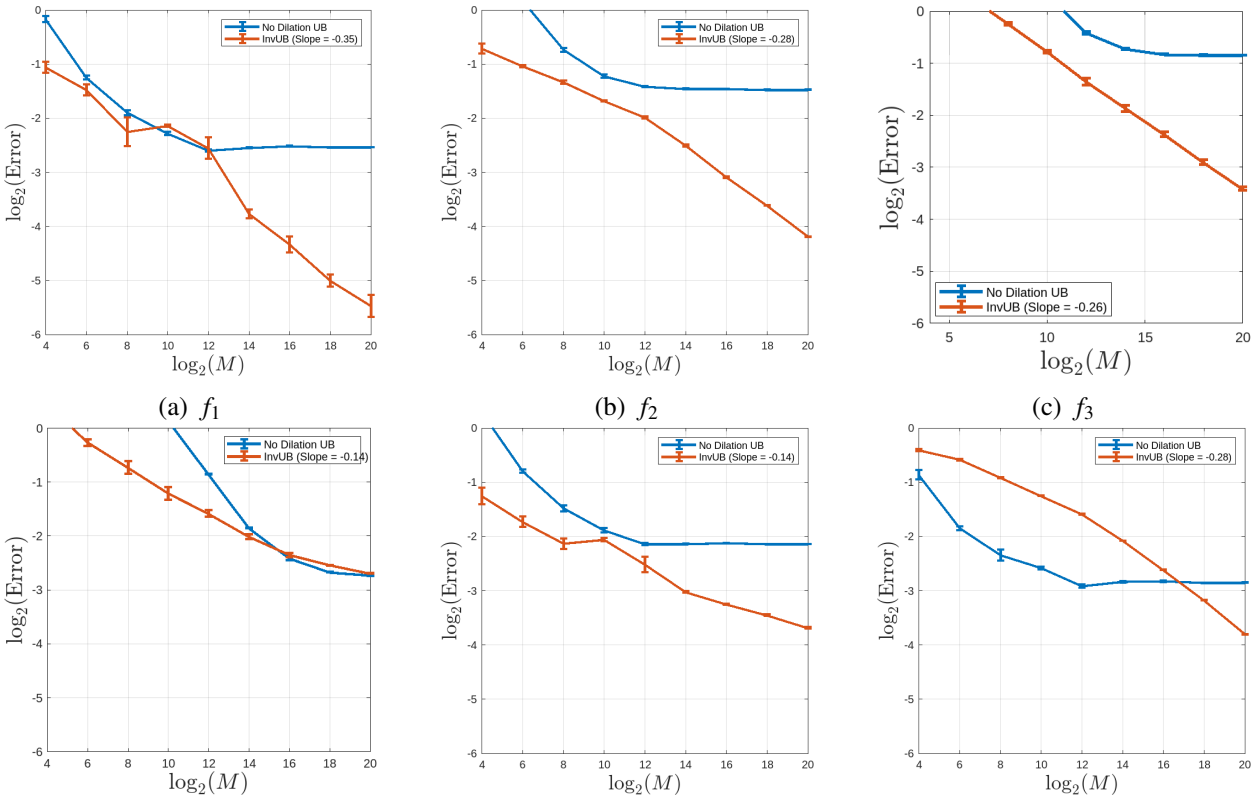
In Figure 2.8, an error plot based on the number of samples is shown with $\sigma = 0.5$; in Figure 2.9, a similar error plot is shown with $\sigma = 1.0$. With regards to the estimation of η, the process given in [26] was unreliable for all sample sizes. Our results show that our modified loss function yields a more reliable estimate of η for large M. See Figures 2.20 and 2.21.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.8 Relative error decay with standard error bars for Bispectrum Recovery using Model 2 without prior knowledge of η and with $\sigma = 0.5$.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.9 Relative error decay with standard error bars for Bispectrum Recovery using Model 2 without prior knowledge of η and with $\sigma = 1.0$.

Our results indicate that we can recovery the bispectrum with a reasonable degree of accuracy. The question is: do we have enough accuracy for full signal inversion?

2.9 Bispectrum Inversion and Hidden Signal Recovery

For hidden signal recovery, the tools we will need are a power spectrum recovery algorithm, a bispectrum recovery algorithm, and a bispectrum inversion algorithm. We use [26] for power spectrum recovery and η estimation, our proposed bispectrum recovery algorithm, and the iterative phase synrchonization algorithm [7] for bispectrum inversion. The general outline of the recovery and inversion process is in Algorithm 2.2. The idea behind algorithm 2.2 is that we have already recovered the magnitudes of the target signal via power spectrum recovery. Thus, if we are able to recover the corresponding phase measurements, will recover the original signal.

The algorithm we use for phase recovery is iterative phase synchronization. Suppose that the

bispectrum of our hidden signal is given by B, and our estimate of the phases for B is given by $\tilde{B}$. Suppose we have an estimate of the phase, $\tilde{y}$, say $\tilde{y}_{k-1}$ that is close to the ground truth. Then we should have $\tilde{B} \circ \overline{T(\tilde{y}_{k-1})} \approx \tilde{y}_{k-1}\tilde{y}_{k-1}^*$, where $T(y)$ is the circulant matrix for the vector y and $\circ$ is the elementwise product of two matrices. We then approximate the phases via solving the optimization problem:

$$\text{argmax}_{z\in\mathbb{C}^n}\text{Re}\{z^*\overline{B \circ T(y)}z\}$$

subject to $|z[\ell]| = 1 \forall \ell$. However, this solution is incorrect by a global phase. We need additional information, namely $\hat{f}(0)$, or some estimate of it.

Algorithm 2.1 describes the iterative phase synchronization algorithm:

Algorithm 2.1 Iterative Phase Synchronization

1: INPUT: normalized bispectrum $\hat{B}$, estimation of phase of $\hat{f}(0)$, given by $\bar{y}(0)$.
2: OUTPUT: Phase of Signal, $\tilde{y}$.
3: Let $k = 0$.
4: **while** Stopping Criterion does not occur **do**
5: Increase k by 1.

$$\hat{y}_k \longleftarrow \text{argmax}_{z\in\mathbb{C}^n}\text{Re}\{z^*\overline{\hat{B} \circ T(\hat{y}_k)}z\} \text{ subject to } z[\ell]| = 1 \quad \forall \ell.$$

6:

$$\hat{y}_k \longleftarrow \hat{y}_k \cdot \frac{\tilde{y}(0)}{\hat{y}(0)}.$$

7: If the signal is real, symmetrize it.
8: **end while**

Other methods, such as frequency marching [7], local non-convex optimization over the manifold of phases [7], and semidefinite relaxation [4]. Frequency marching and iterative phase synchronization have been found the have similar empirical performance. However, iterative phase synchronization requires less assumptions, namely one does not need $\hat{f}(1) \neq 0$. We have found that local non-convex optimization over the manifold of phases did not yield good results in our experiments, and semidefinite programming was infeasible due to memory requirements.

Algorithm 2.2 Hidden Signal Recovery Algorithm

1: INPUTS: noisy signals $\{y_j\}$
2: Calculate $\tilde{f} = \sum_j \hat{y}_j, \tilde{p}_\eta + \tilde{p}_\sigma, \tilde{g}_\eta + \tilde{g}_\sigma$.
3: Estimate the additive noise level, $\tilde{\sigma}$, using the power spectrum on the edge of the signal.
4: Perform additive noise centering on the power spectrum and bispectrum.
5: **if** Eta Known **then**
6: Recover the power spectrum via solving the convex optimization problem $\text{argmin}_{\tilde{p}} \|(I - S_{C_0})\tilde{p} - C_1 S_{C_2} p_{\text{data}}\|_2^2$.
7: **else if** Eta Unknown **then**
8: Estimate the power spectrum and η via solving the nonconvex optimization problem $\text{argmin}_{\tilde{p},\tilde{\eta}} \frac{\|\left(I - S_{C_0(\tilde{\eta})}\right)\tilde{p} - C_1(\tilde{\eta}) S_{C_2(\tilde{\eta})} p_{\text{data}}\|_2^2}{\eta^2}$.
9: **end if**
10: Recover the bispectrum via solving the convex optimization problem $\text{argmin}_{\tilde{g}} \|(I - L_{C_0})\tilde{g} - C_1 L_{C_2} d\|_2^2$.
11: Apply APS to recover the original signal using $\tilde{f}(0)$, the recovered power spectrum, and the recovered bispectrum.

To illustrate the difficulty of this process, we provide a ground truth example for f_5 in Figure 2.10 and four observations of f_5 in Figure 2.11. One can see that the observations do not resemble the actual signal, even at a low noise levels.

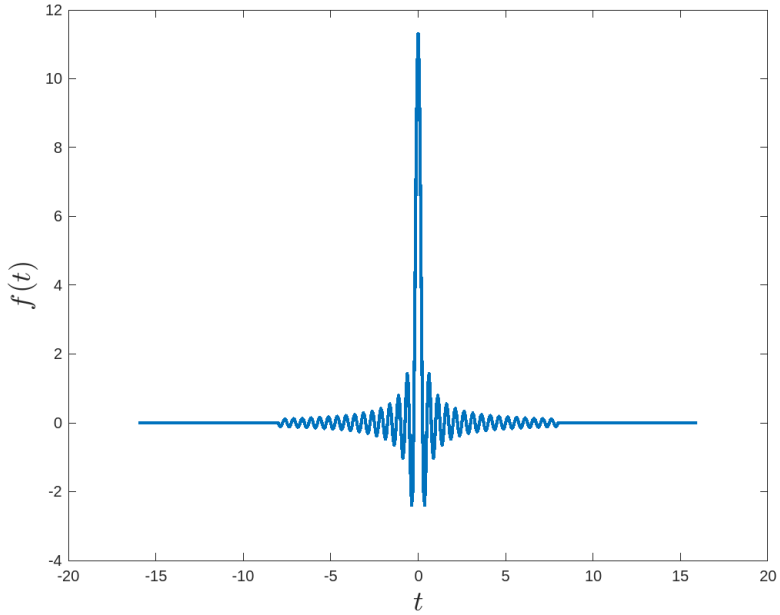
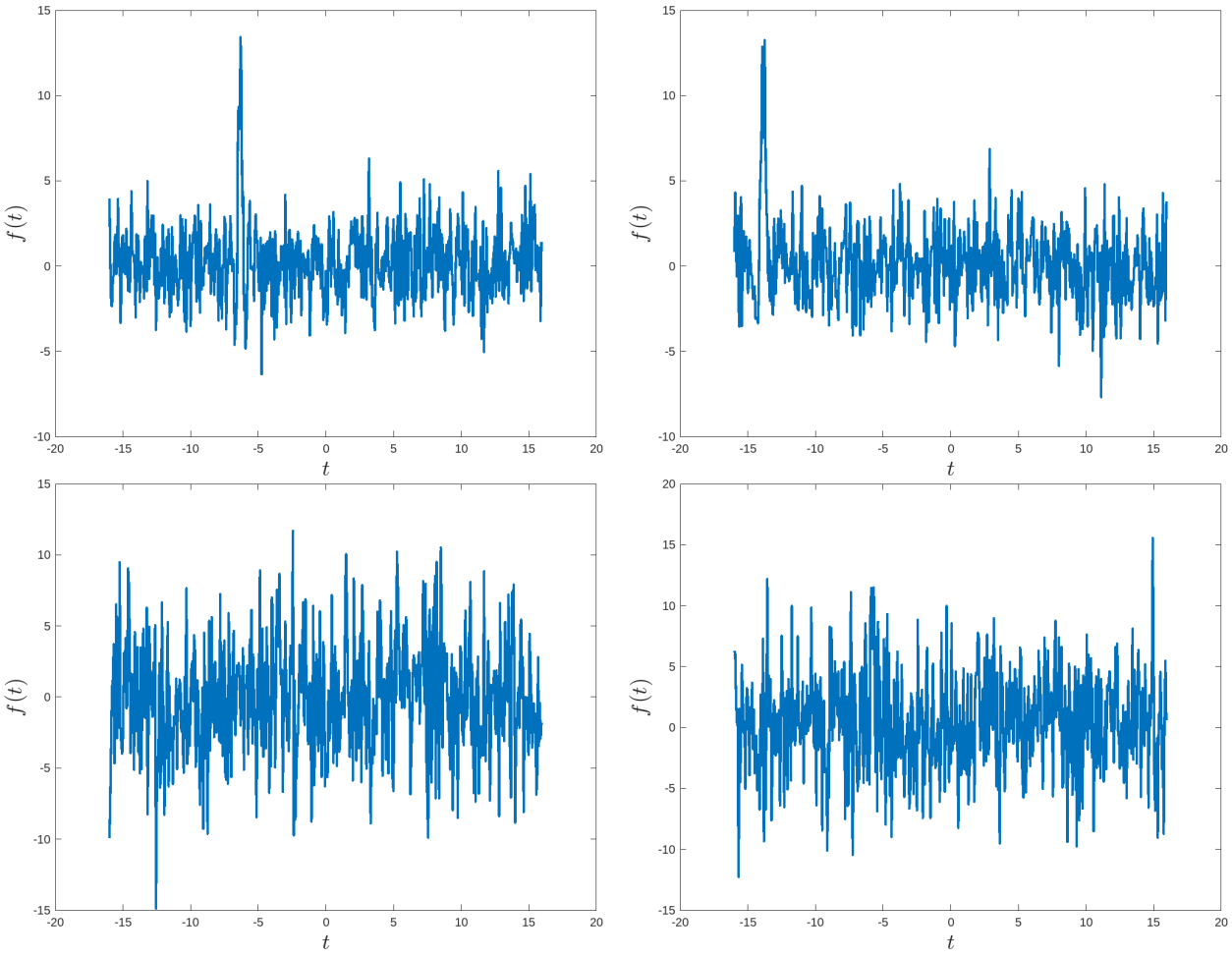


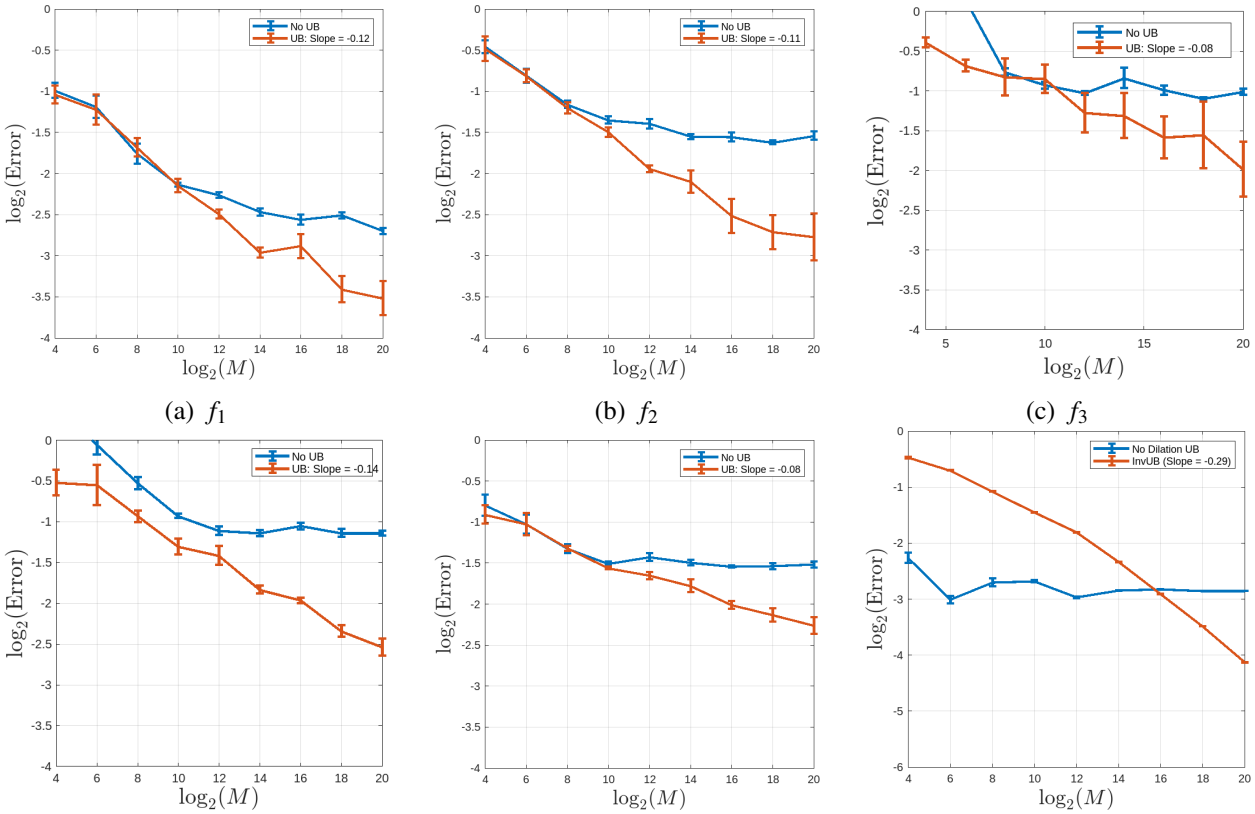
Figure 2.10 Ground Truth Signal for f_5.



(a) f_3

Figure 2.11 Corrupted Samples for f_5. The top row is $\sigma = 0.5$ and the bottom row is $\sigma = 1.0$.

We test our results on the signals from the previous section with $M = 2^{20}$ samples with $\eta = 12^{-1/2}$. We start with an oracle η. The results for $\sigma = 0.5$ are given in Figure 2.12 and example inversion plots are given in Figure 2.13.

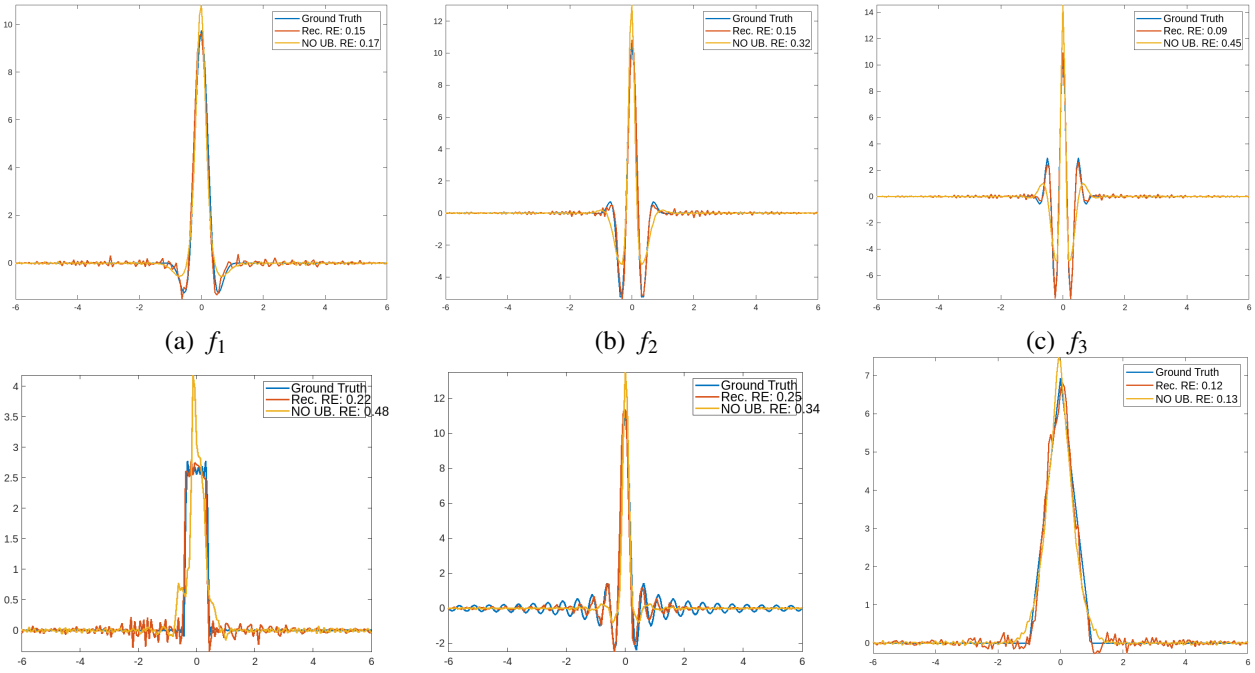


(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.12 Relative error decay with standard error bars for bispectrum inversion using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 0.5$.

The blue lines are for inversion using additive noise centered versions of the average bispectrum and average power spectrum, and the red lines are using our inversion unbiasing procedure. One can see that there is a huge gain in performance for the first five signals. However, for f_6, there is no gain in performance. We hypothesize this occurs because the power spectrum and bispectrum of f_6 have a peak around the origin. Dilating the power spectrum and bispectrum will not have a large effect on the shape of the signal, since their support is restricted to very low frequencies. This behavior is observed in Figure 2.7 as well.



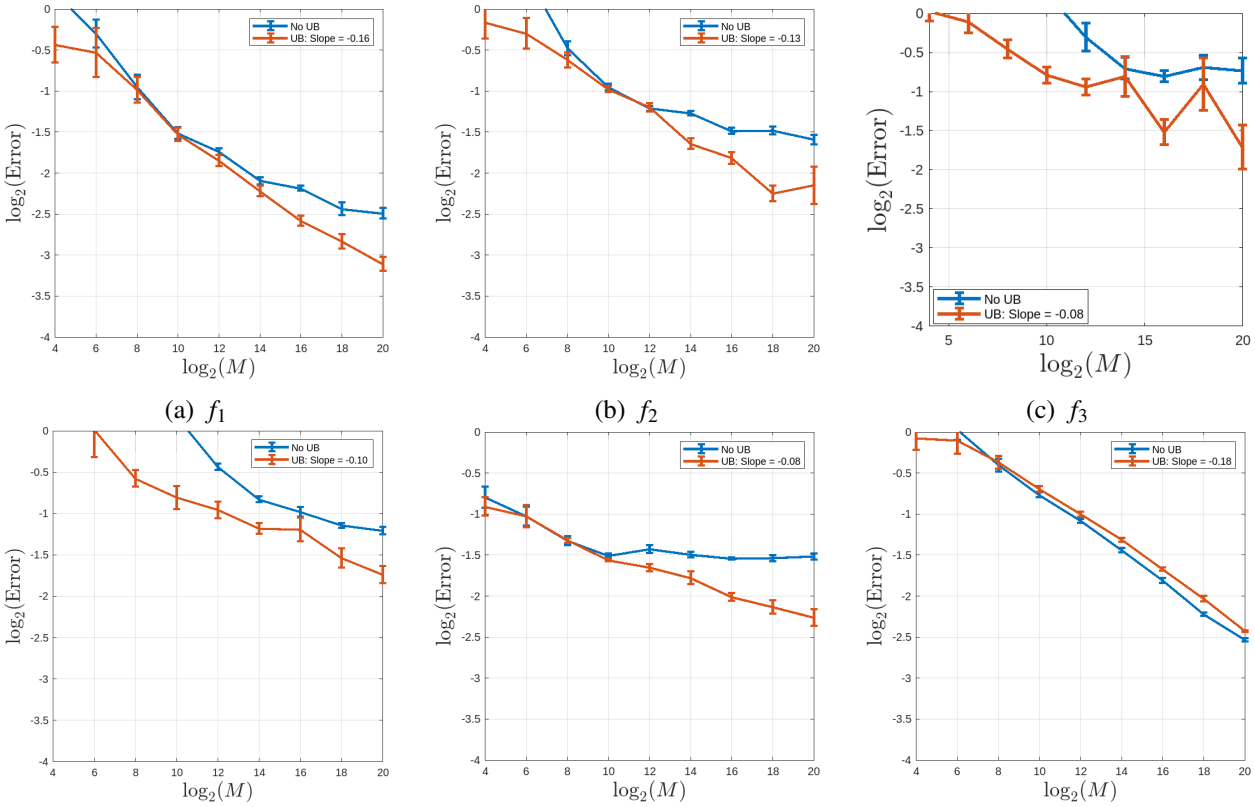
(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.13 Bispectrum inversion results using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 0.5$. "Rec" is using our inversion unbiasing procedure and "NO UB" is using the the centered averages. "RE" stands for the relative error.

Note that while the relative error between the signals is similar in a few cases, such as f_1 and f_5, the quality of the recovery results cannot be fully judged by using relative error. For instance, in f_1 and f_5, one can see that the general shape of the signals is incorrect without the inversion unbiasing, but the relative error is low because the recovered signal is relatively smooth. On the other hand, with the inversion unbiasing, the recovered signal has the right general shape, but is not very smooth. Also, with regards to f_5, the assumptions of the model are not met, so the drop in performance is expected. For this specific run, for the high frequency signal (f_3), the recovery error decreased from 0.45 without unbiasing to 0.09 with unbiasing. Additionally for f_2 and f_4, the unbiasing procedure decreased the error error by over 50%. For f_1 and f_6 we do not see much improvement, but that is as expected since they are supported in the low frequencies; for f_5 there is a moderate improvement, but the bispectrum estimation was less reliable because model assumptions were not met.

Now we consider the case where $\sigma = 1.0$. The error decay plots are given in Figure 2.14 and corresponding inversion plots are given in Figure 2.15.

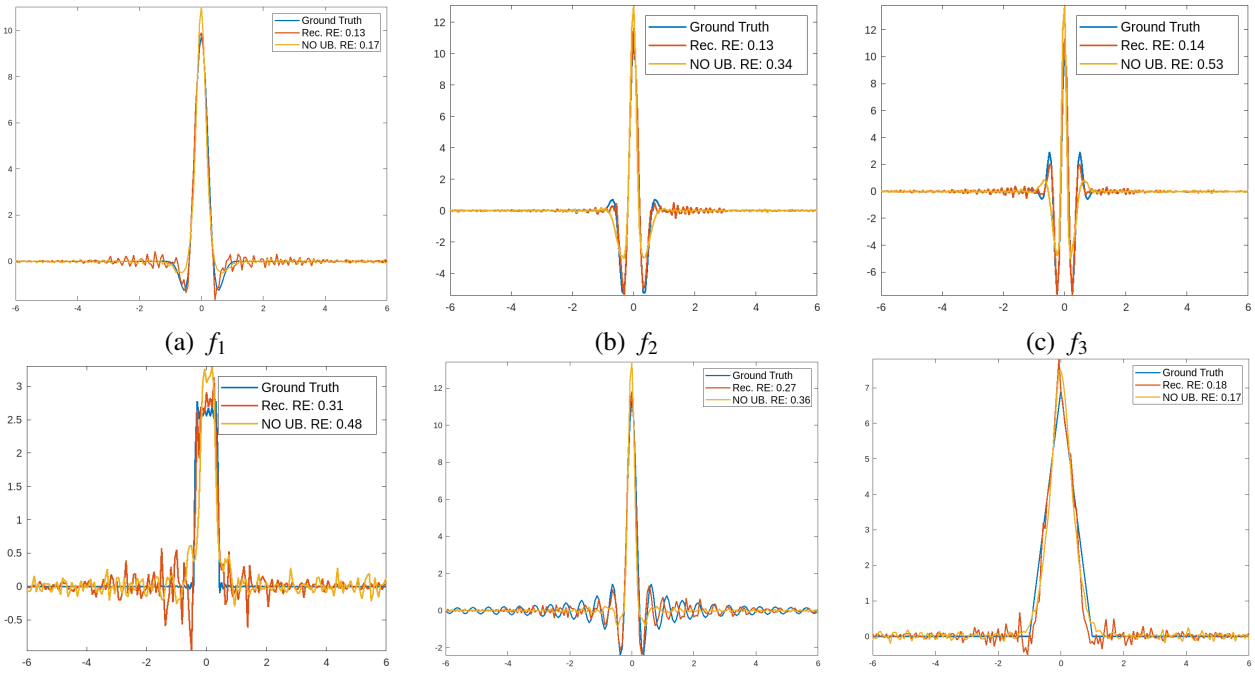


(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.14 Relative error decay with standard error bars for bispectrum inversion using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 1.0$.

Notice that the gap in relative error between not using the unbiasing procedure and using the unbaising procedure has decreased. This suggests that the bispectrum inversion algorithm we have used is sensitive to noise. This also explains the slight performance drop we have observed.



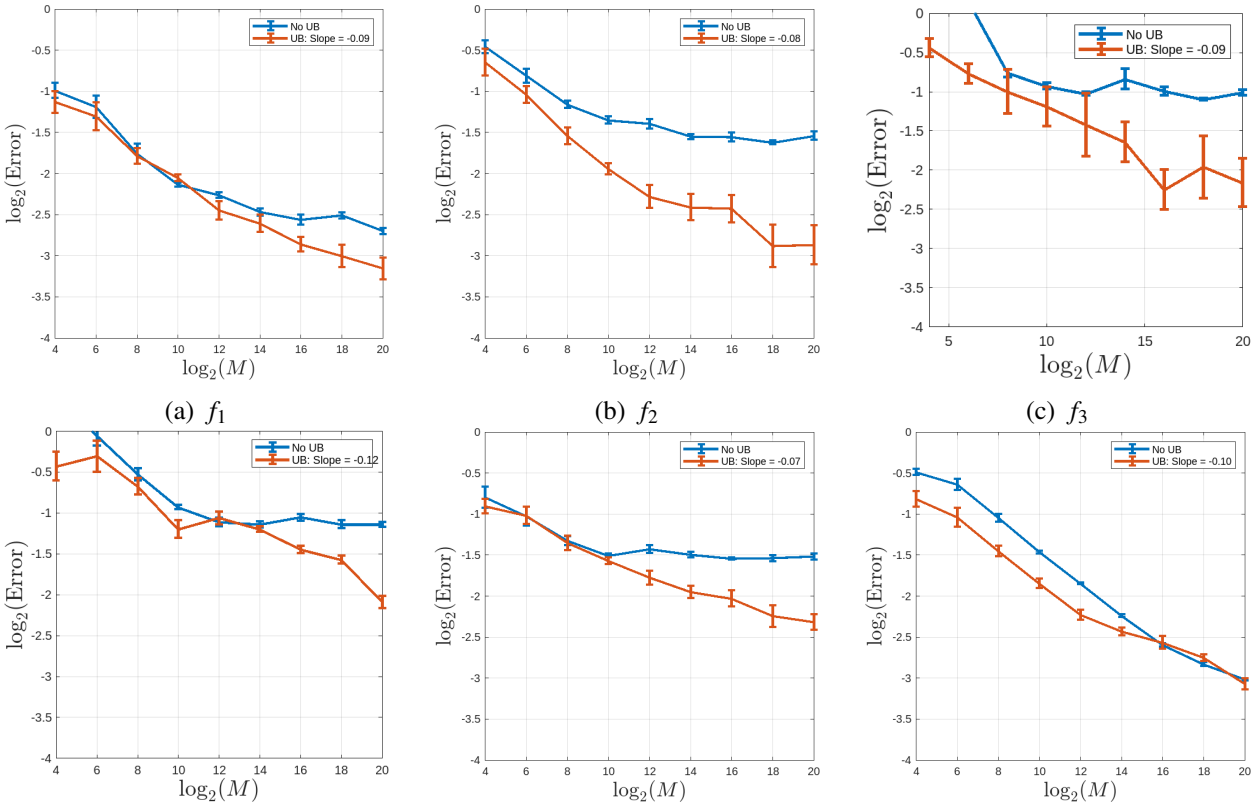
(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.15 Bispectrum inversion results using Model 2 under the assumption of oracle $\eta = 12^{-1/2}$ with $\sigma = 1.0$.

Surprisingly, our error has not increased too much except in f_4. For f_4, we notice that noise in the signal is most notable around the discontinuities of the function. There is a similar phenomenon in f_6. At the end of each triangle and the peak of the triangle, we observe more noise. This suggests that the bispectrum inversion algorithm has a hard time learning discontinuities of the signal, which is expected.

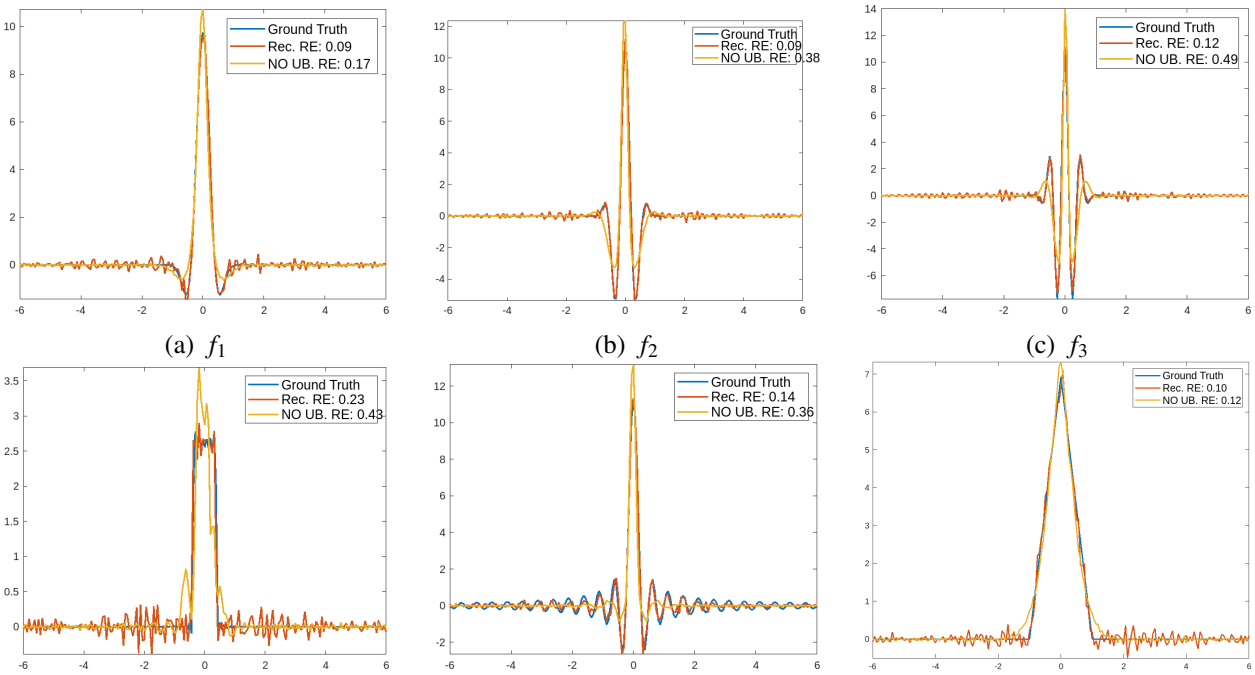
Now, we consider the case where η is unknown and estimated again. For $\sigma = 0.5$, the error decay plots are in Figure 2.16 and the inversion results are in Figure 2.17. For $\sigma = 1.0$, the error decay plots are in Figure 2.18 and the inversion results are in Figure 2.19. Surprisingly, the results are similar to the corresponding oracle plots in many cases.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

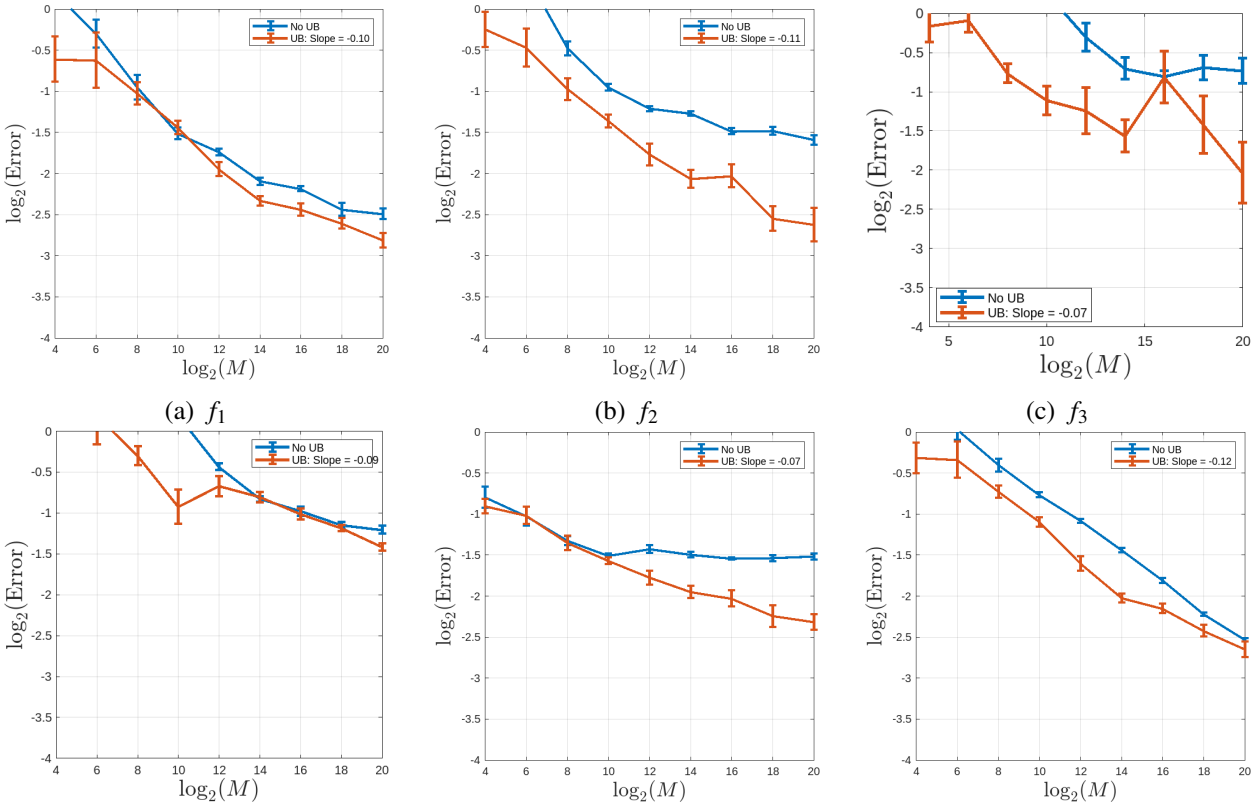
Figure 2.16 Relative error decay with standard error bars for bispectrum inversion using Model 2 with $\sigma = 0.5$ and no prior knowledge of η.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

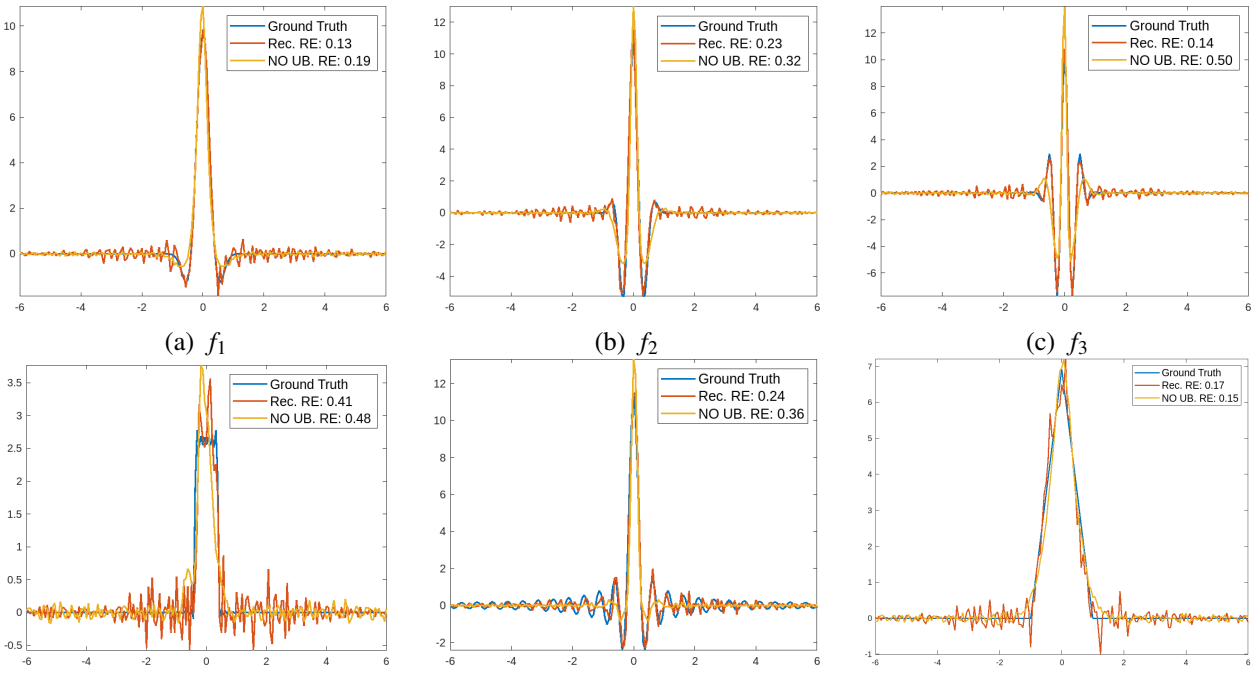
Figure 2.17 Bispectrum inversion results using Model 2 with $\sigma = 0.5$ and no prior knowledge of η.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.18 Relative error decay with standard error bars for bispectrum inversion using Model 2 with $\sigma = 1.0$ and no prior knowledge of η.

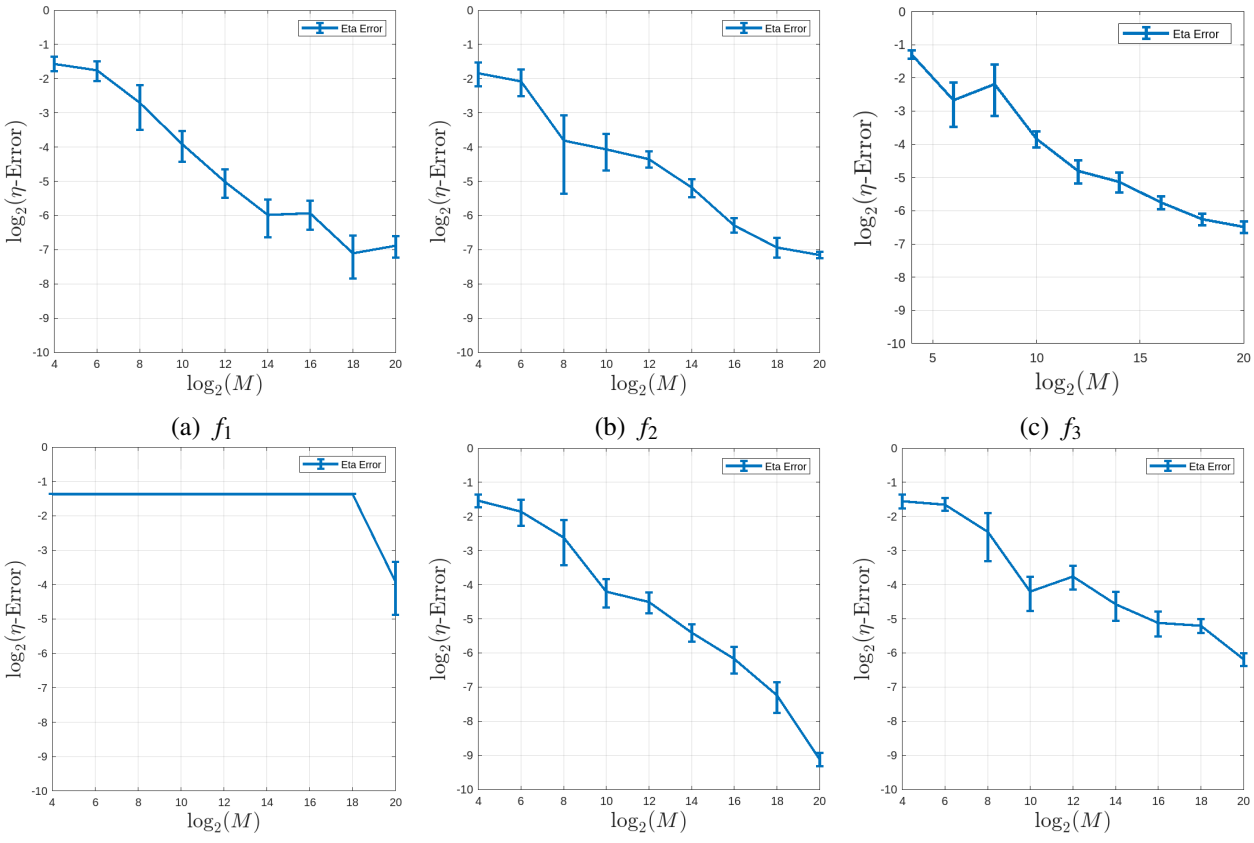


(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.19 Bispectrum inversion results using Model 2 with $\sigma = 1.0$ and no prior knowledge of η.

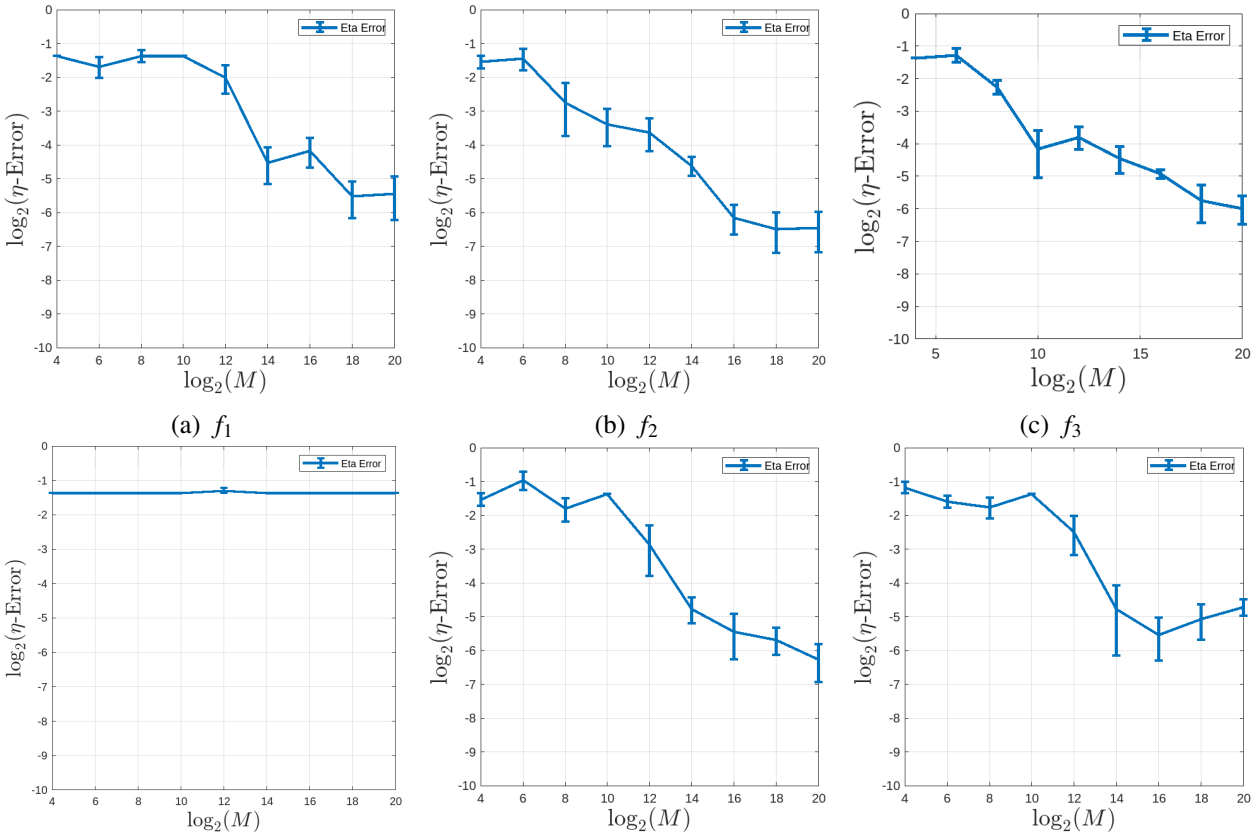
Finally, here are error plots for the estimation of η in the empirical case. Figure 2.20 is for $\sigma = 0.5$ and Figure 2.21 is for $\sigma = 1.0$. Note that the improvement tends to plateau once we have more than 2^{14} samples for most of our cases. Note that the η estimation was subpar for f_4. We believe this is because the signal is not continuous, but we have not done rigorous tests to confirm this. Nonetheless, the bispectrum recovery results were still adequate in the oracle case.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.20 Mean Relative Error in estimating η with $\sigma = 0.5$.



(a) f_1 (b) f_2 (c) f_3

(d) f_4 (e) f_5 (f) f_6

Figure 2.21 Mean Relative Error in estimating η with $\sigma = 1.0$.

2.10 Conclusions and Future Work

In this chapter, we have provided a solution for the dilation MRA model by recovering the bispectrum of the noisy observations. After recovering the bispectrum, we perform full signal inversion and observe competitive results compared to other methods without dilations.

One natural extension is to work in two dimensions. That is, we want to recover a signal $f : \mathbb{R}^2 \to \mathbb{R}$ from many noisy observations that have been randomly translated, dilated, **rotated**, and corrupted by additive noise. Define the rotation matrix

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{2.22}$$

A formal description is now given by

Model 4. Suppose we have M independent observations of a function $f \in L^2(\mathbb{R}^2)$ defined by

$$y_j(x) = f(R_{\theta_j}^{-1}(1-\tau_j)^{-1}(x-t_j)) + \varepsilon_j(x) := f_j(x) + \varepsilon_j(x), \quad 1 \le j \le M$$

Furthermore, assume that

- $\operatorname{supp}(f_j) \subset [-\frac{1}{2}, \frac{1}{2}]^2$ for $1 \le j \le M$.
- $\{t_j\}_{j=1}^M$ are independent samples of a random variable $t \in \mathbb{R}$.
- $\{\theta_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\theta \in [-\pi, \pi)$.
- $\{\tau_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\tau \in \mathbb{R}$ satisfying $\mathbb{E}[\tau] = 0$ and $\operatorname{Var}(\tau) = \eta^2 \le \frac{1}{12}$.
- $\{\varepsilon_j(x)\}_{j=1}^M$ are independent white noise processes on $[-\frac{1}{2}, \frac{1}{2}]^2$ with variance σ^2.

Can we recover f?

The main difficulty in this model comes from the addition of the rotations, like mentioned in previous sections. Even the noiseless case itself is difficult because one cannot simply align the signals anymore. Intuitively, one has to first rotate all the signals in place, but this is not very feasible. The corresponding noiseless model is given by

Model 5. Suppose we have M independent observations of a function $f \in L^2(\mathbb{R}^2)$ defined by

$$f_j(x) = f(R_{\theta_j}^{-1}(1-\tau_j)^{-1}(x-t_j)) \quad 1 \le j \le M$$

Furthermore, assume that

- $\text{supp}(f_j) \subset [-\frac{1}{2}, \frac{1}{2}]^2$ for $1 \leq j \leq M$.
- $\{t_j\}_{j=1}^M$ are independent samples of a random variable $t \in \mathbb{R}$.
- $\{\theta_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\theta \in [-\pi, \pi)$.
- $\{\tau_j\}_{j=1}^M$ are independent samples of a uniformly distributed random variable $\tau \in \mathbb{R}$ satisfying $\mathbb{E}[\tau] = 0$ and $\text{Var}(\tau) = \eta^2 \leq \frac{1}{12}$.

Can we recover f?

We will start with Model 4 and generalize our results to Model 5, like in the one dimensional case.

CHAPTER 3

NONLINEAR HEEGER-BERGEN TEXTURE SYNTHESIS

3.1 Background on Texture Synthesis

Texture synthesis is the process of generating an image from a reference image by taking advantage of its statistical properties. The new texture should have similar qualities as the reference texture, but look different. In other words, for example, the same reference texture, a rotation of the reference texture, or a rearrangement of the reference texture should not be the result after synthesis. An example of texture synthesis is given in Figure 3.1.

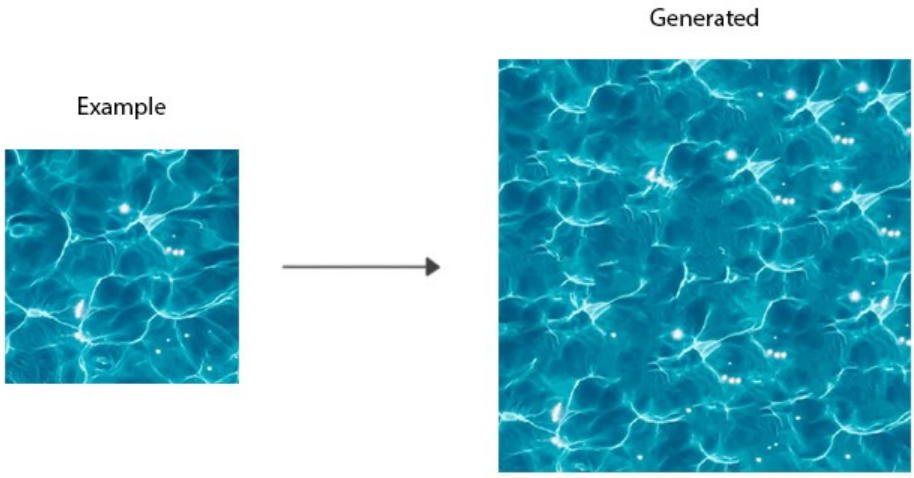


Figure 3.1 "Example" is the reference texture and "Generated" is the synthesized texture. Note that the images have the same perceptual qualities, but are not the same image.

While this task can be phrased simply, one difficulty in this task is that traditional image metrics do not lead to good synthesis results. For example, if one were to minimize the MSE loss between a reference image I_R and synthesized image I_S via solving the optimization problem

$$I_S = \text{argmin}_{\hat{I}_S} \|\hat{I}_S - I_R\|_2^2,$$

the result would be same texture. To illustrate why such a metric is difficult, we will provide some examples.

Consider Figure 3.2. The synthesis in the last two columns have similar quality compared to the reference, and it's clear that both textures are not simply repetitions of the reference texture.

Figure 3.2 **Left:** Original Texture. Middle: Synthesis using [24]. **Right:** Synthesis using [20].

One could argue that dots are more solid in the middle column, which matches with the reference texture, so the synthesis is better. However, this is a subjective argument without any quantitative backing.

Additionally, consider Figure 3.3. One can see that the middle column has worse alignment compared to the reference texture on the left, so it's clear that the synthesis on the right is higher quality since there is no repetition.

Figure 3.3 **Left:** Original Texture. Middle: Synthesis using [24]. **Right:** Synthesis using [20].

The synthesis in the right column of Figure 3.3 captures what we will call "long range constraints" in an image. That is, it is able to capture macroscopic features that the middle column of Figure 3.3 cannot capture. To approach this problem, there are two factors to consider. First, can one find a representation that captures texture? If so, what is a good measure for measuring similarity between textures? One needs to account for long range constraints without creating a

texture too similar to the original reference texture.

One approach, which we will generalize in this chapter, is to update a white noise by matching the histograms of wavelet coefficients between a reference image and the white noise. First, we will need to introduce the filter bank we used and the wavelet transform.

3.2 Filter Construction

In this section we describe the filter bank used in the Heeger-Bergen algorithm. This is the analytic description of the filters that does not rely on down sampling. We are following the presentation in [10], but some of the notation is changed so care should be taken when comparing the two. All filters are defined in frequency in the frequency box $[-\pi, \pi]^2 := [-\pi, \pi] \times [-\pi, \pi]$ using polar coordinates.

Write $\omega = (\omega_1, \omega_2) \in [-\pi, \pi)^2$ as

$$\begin{aligned} \omega &= (r, \theta) \\ r &:= \sqrt{\omega_1^2 + \omega_2^2} \\ \theta &:= \begin{cases} \pi & \omega_1 \leq 0 \text{ and } \omega_2 = 0 \\ 2 \arctan\left(\frac{\omega_2}{\omega_1 + r}\right) & \text{otherwise} \end{cases} . \end{aligned}$$

We begin by defining some "building block" functions. The first is a low frequency radial function $L : [0, \infty) \to \mathbb{R}$ defined as

$$\forall\, r \geq 0, \quad L(r) := \begin{cases} 1 & r \leq \pi/2 \\ \cos\left(\frac{\pi}{2} \log_2\left(\frac{2r}{\pi}\right)\right) & \pi/2 < r \leq \pi \\ 0 & r \geq \pi \end{cases} .$$

The second is a high frequency function $H : [0, \infty) \to \mathbb{R}$ defined as

$$\forall\, r \geq 0, \quad H(r) := \begin{cases} 0 & r \leq \pi/2 \\ \cos\left(\frac{\pi}{2} \log_2\left(\frac{r}{\pi}\right)\right) & \pi/2 < r \leq \pi \\ 1 & r \geq \pi \end{cases} .$$

One can verify that L and H satisfy the following important property:

$$|L(r)|^2 + |H(r)|^2 = 1, \quad \forall\, r \geq 0. \tag{3.1}$$

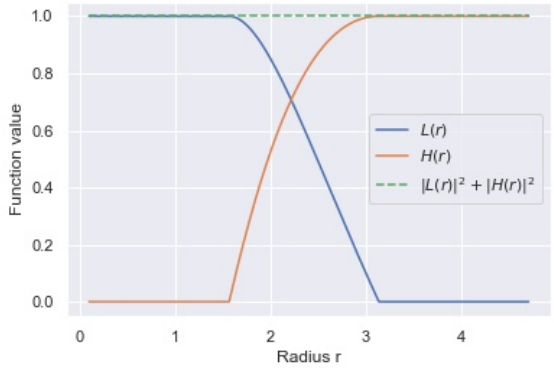


Figure 3.4 Plot of $L(r)$, $H(r)$, and $|L(r)|^2 + |H(r)|^2$.

In fact, the cosine part for $L(r)$ can be replaced with a different decreasing function, and the cosine part of $H(r)$ can be replaced with a different increasing function, so long as (3.1) holds. Figure 3.4 plots $L(r)$ and $H(r)$ and verifies, numerically, equation (3.1).

Remark 1. Looking at the plots in Figure 3.4, indeed we may want to use a different form for $L(r)$ and $H(r)$, as they are not very smooth at the point where they equal zero.

We also define dilations of L and H. They are given by:

$$\forall\, j \in \mathbb{Z}, \quad L_j(r) := L(2^j r) = \begin{cases} 1 & r \le \pi/2^{j+1} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2^{j+1}r}{\pi}\right)\right) & \pi/2^{j+1} < r \le \pi/2^j \\ 0 & r \ge \pi/2^j \end{cases},$$

and

$$\forall\, j \in \mathbb{Z}, \quad H_j(r) := H(2^j r) = \begin{cases} 0 & r \le \pi/2^{j+1} \\ \cos\left(\frac{\pi}{2}\log_2\left(\frac{2^{j}r}{\pi}\right)\right) & \pi/2^{j+1} < r \le \pi/2^j \\ 1 & r \ge \pi/2^j \end{cases}.$$

We remark that it follows from (3.1) that

$$|L_j(r)|^2 + |H_j(r)|^2 = 1, \quad \forall\, r \ge 0, \ \ \forall\, j \in \mathbb{Z}.$$

Now we define a family of functions for the angular variable θ. Let Q be the number of such functions, which will later correspond to the number of directional filters at each scale. We define

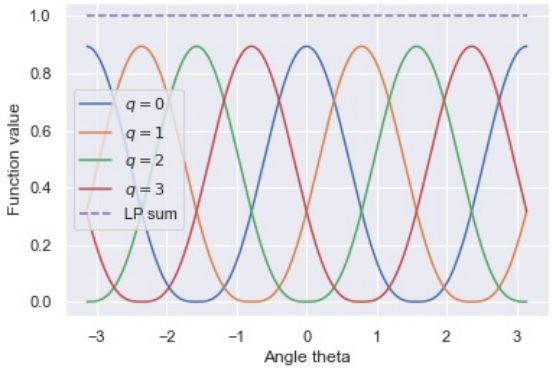


Figure 3.5 Plots of $G_q(\theta)$ and $\sum_{q=0}^{Q-1} |G_q(\theta)|^2$ for $0 \le q < Q = 4$.

$\forall\, 0 \le q < Q,\ \ \forall\, \theta \in (-\pi, \pi]$,

$$G_q(\theta) := \alpha_Q \left(\cos\left(\theta - \frac{\pi q}{Q}\right)^{Q-1} \mathbb{1}_{|\theta - \pi q/Q| \le \pi/2} + \cos\left(\theta - \frac{\pi(q-Q)}{Q}\right)^{Q-1} \mathbb{1}_{|\theta - \pi(q-Q)/Q| \le \pi/2} \right),$$

where

$$\alpha_Q := 2^{Q-1} \frac{(Q-1)!}{\sqrt{Q(2(Q-1))!}}.$$

One can verify that

$$\sum_{q=0}^{Q-1} |G_q(\theta)|^2 = 1, \quad \forall\, \theta \in (-\pi, \pi]. \tag{3.2}$$

Figure 3.5 plots the functions $G_q(\theta)$ and verifies (3.2) numerically. Similarly to the $L(r)$ and $H(r)$ functions, the functions $G_q(\theta)$ can be changed so long as (3.2) is satisfied.

Now we use the functions $L(r)$, $H(r)$, and $G_q(\theta)$ to build our filters. We construct three types of filters:

- A high frequency filter $h : \mathbb{R}^2 \to \mathbb{R}$
- Directional wavelet filters $\psi_q : \mathbb{R}^2 \to \mathbb{R}$
- A low frequency filter $\ell : \mathbb{R}^2 \to \mathbb{R}$

The high pass filter $h(u)$ is defined through its Fourier transform $\widehat{h}(r, \theta)$ as

$$\forall\, r \ge 0,\ \ \forall\, \theta \in (-\pi, \pi], \quad \widehat{h}(r, \theta) := H(r),$$

and the low pass filter $\ell(u)$ is defined analogously,

$$\forall\, r \ge 0,\ \ \forall\, \theta \in (-\pi, \pi], \quad \widehat{\ell}(r, \theta) := L(r).$$

The directional filters $\psi_q(u)$ are defined as

$$\forall\, r \geq 0,\ \ \forall\, \theta \in (-\pi, \pi], \quad \widehat{\psi}_q(r,\theta) := L_0(r) H_1(r) G_q(\theta).$$

One may verify that all the filters are symmetric through the origin (i.e., $h(-u) = h(u)$, $\ell(-u) = \ell(u)$, and $\psi_q(-u) = \psi_q(u)$) and real valued in space.

We now define the wavelet transform of an image $x \in \mathbf{L}^2(\mathbb{R}^2)$ with its frequency support contained in $[-\pi, \pi]^2$, i.e., $\mathrm{supp}(\widehat{x}) \subseteq [-\pi, \pi]^2$. Let $J \geq 1$ be the number of scales and let $Q \geq 1$ be the number of orientations. A dilation of a filter $f : \mathbb{R}^2 \to \mathbb{C}$ is defined as

$$\forall\, u \in \mathbb{R}^2,\ \ \forall\, j \in \mathbb{Z}, \quad f_j(u) := 2^{-2j} f(2^{-j} u).$$

We remark that the Fourier transform of a dilated filter satisfies:

$$\widehat{f_j}(r,\theta) = \widehat{f_j}(\omega) = \widehat{f}(2^j \omega) = \widehat{f}(2^j r, \theta), \quad \forall\, \omega \in \mathbb{R}^2,\ \ \forall\, j \in \mathbb{Z}.$$

Using this notation define the filter bank $\mathcal{F}_{J,Q}$ as

$$\mathcal{F}_{J,Q} := \{h\,,\, \ell_J\,,\, \psi_{j,q} : 0 \leq j < J,\ \ 0 \leq q < Q\},$$

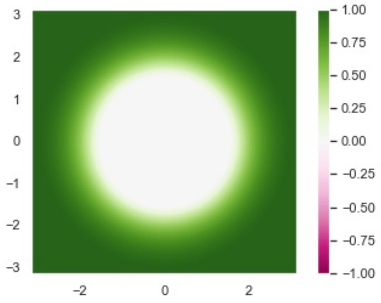
where $\ell_J(u)$ is the dilation of the low pass filter $\ell(u)$, which means:

$$\ell_J(u) = 2^{-2J} \ell(2^{-J} u) \quad \Longrightarrow \quad \widehat{\ell}_J(r,\theta) = \widehat{\ell}(2^J r, \theta) = L(2^J r) = L_J(r),$$

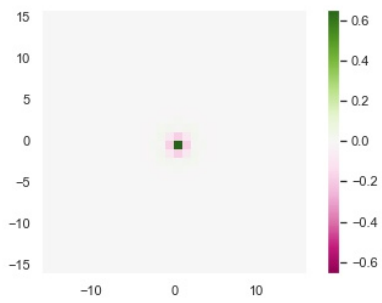
and $\psi_{j,q}(u)$ is the dilation of the directional filter $\psi_q(u)$, meaning that:

$$\psi_{j,q}(u) = 2^{-2j} \psi_q(2^{-j} u) \quad \Longrightarrow \quad \widehat{\psi}_{j,q}(r,\theta) = \widehat{\psi}_q(2^j r, \theta) = L_j(r) H_{j+1}(r) G_q(\theta).$$

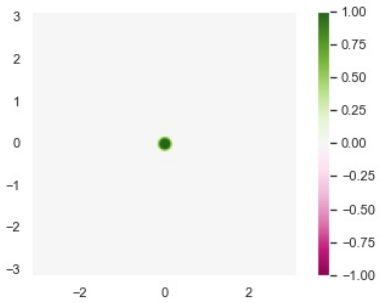
Figure 3.6 plots the high frequency residual filter $h(u)$ and the low frequency filter $\ell_J(u)$ for $J = 4$, along with their Fourier transforms. Figure 3.7 plots the Fourier transforms $\widehat{\psi}_{j,q}(\omega)$ of the directional band-pass wavelets for $0 \leq j < J = 4$ and $0 \leq q < Q = 4$, while Figure 3.8 plots $\psi_{j,q}(u)$ in space for $0 \leq j < J = 4$ and $0 \leq q < Q = 4$.



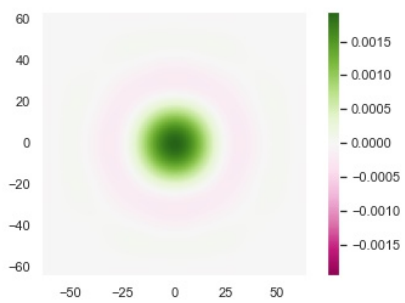
(a) $\widehat{h}(\omega)$



(b) $h(u)$



(c) $\widehat{\ell}_J(\omega)$ for $J = 4$



(d) $\ell_J(u)$ for $J = 4$

Figure 3.6 The high frequency residual filter $h(u)$ and the low frequency filter $\ell_J(u)$ with $J = 4$, and their Fourier transforms $\widehat{h}(\omega)$ and $\widehat{\ell}_J(\omega)$, respectively.

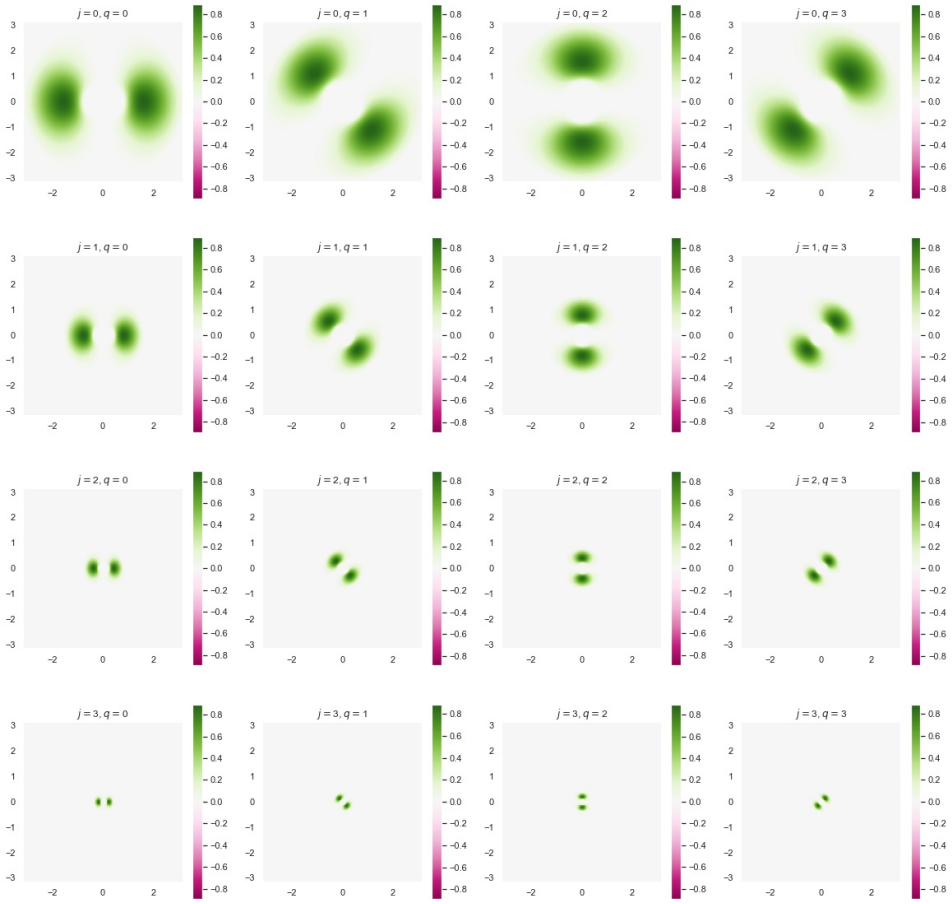


Figure 3.7 The Fourier transforms $\widehat{\psi}_{j,q}(\omega)$ of the directional wavelets for $0 \leq j < J = 4$ and $0 \leq q < Q = 4$.

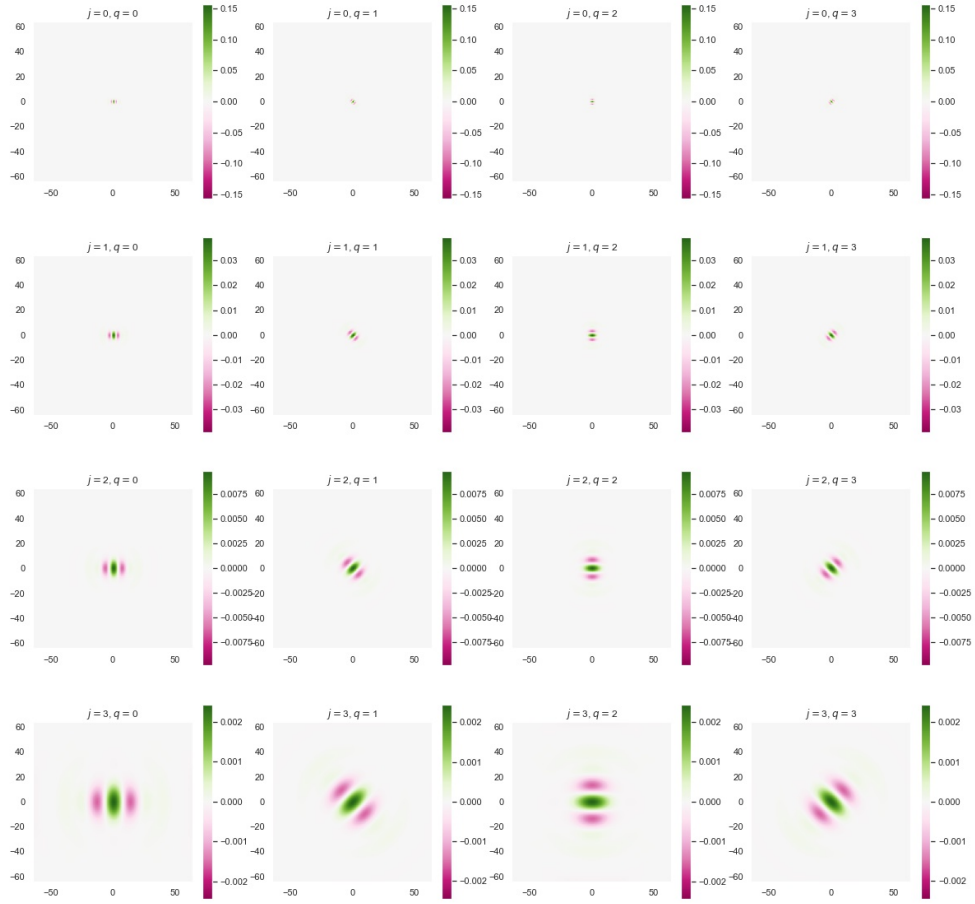


Figure 3.8 The directional wavelets $\psi_{j,q}(u)$ for $0 \leq j < J = 4$ and $0 \leq q < Q = 4$.

Define the wavelet transform of $x(u)$ as the map $W_{J,Q} : \mathbf{L}^2(\mathbb{R}^2) \to \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$, where

$$W_{J,Q}x := \{x * h\,,\; x * \ell_J\,,\; x * \psi_{j,q} : 0 \le j < J,\;\; 0 \le q < Q\}.$$

Notice that $W_{J,Q}$ takes an image $x(u)$ and returns $JQ + 2$ images, which we refer to as the wavelet coefficients of x. Let $\|x\|$ denote the $\mathbf{L}^2(\mathbb{R}^2)$ norm of x,

$$\|x\|^2 = \int_{\mathbb{R}} |x(u)|^2\, du.$$

The norm of $W_{J,Q}x$ is defined as

$$\|W_{J,Q}x\|^2 := \|x * h\|^2 + \|x * \ell_J\|^2 + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} \|x * \psi_{j,q}\|^2.$$

This particular wavelet transform has many nice properties due to the filter construction. We collect them in the next theorem.

Theorem 8. The filter bank $\mathcal{F}_{J,Q}$ satisfies the following Littlewood-Paley condition:

$$|\widehat{h}(\omega)|^2 + |\widehat{\ell_J}(\omega)|^2 + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} |\widehat{\psi}_{j,q}(\omega)|^2 = 1, \quad \forall\, \omega \in [-\pi, \pi]^2. \tag{3.3}$$

Therefore, for any $x \in \mathbf{L}^2(\mathbb{R}^2)$ with $\mathrm{supp}(\widehat{x}) \subseteq [-\pi, \pi]^2$, the wavelet transform $W_{J,Q}$ is an isometry,

$$\|W_{J,Q}x\| = \|x\|,$$

and furthermore has the following inverse:

$$x = x * h * h + x * \ell_J * \ell_J + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} x * \psi_{j,q} * \psi_{j,q}. \tag{3.4}$$

Proof. First, convert everything to polar coordinates. Use condition (3.2) to simplify the third term, condition (3.1) for the other terms, and also use the definition of $L_j(r)$ to observe that

$L_j(r)L_{j+1}(r) = L_{j+1}(r)$, which results in a telescoping sum:

$$
\begin{aligned}
&|\widehat{h}(\omega)|^2 + |\widehat{\ell_J}(\omega)|^2 + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} |\widehat{\psi}_{j,q}(\omega)|^2 \\
&= |H(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} |L_j(r)|^2 |H_{j+1}(r)|^2 |G_q(\theta)|^2 \\
&= |H(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1} |L_j(r)|^2 |H_{j+1}(r)|^2 \left| \left(\sum_{q=0}^{Q-1} G_q(\theta) \right)^2 \right| \\
&= |H(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1} |L_j(r)|^2 |H_{j+1}(r)|^2 \\
&= 1 - |L(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1} |L_j(r)|^2 (1 - |L_{j+1}(r)|^2) \\
&= 1 - |L(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1} |L_j(r)|^2 - |L_j(r)|^2 |L_{j+1}(r)|^2 \\
&= 1 - |L(r)|^2 + |L_J(r)|^2 + \sum_{j=0}^{J-1} |L_j(r)|^2 - |L_{j+1}(r)|^2 \\
&= 1 - |L(r)|^2 + |L_J(r)|^2 + |L(r)|^2 - |L_J(r)|^2 \\
&= 1.
\end{aligned}
$$

To prove the isometry property, multiply both sides of (3.3) by $|\hat{x}(\omega)|^2$, integrate, and apply Plancherel formula.

To prove the inversion formula, take the Fourier Transform of

$$g = x * h * h + x * \ell_J * \ell_J + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} x * \psi_{j,q} * \psi_{j,q}.$$

Then

$$\begin{aligned}\hat{g}(\omega) &= \hat{x}(\omega)\hat{h}(\omega)\hat{h}(\omega) + \hat{x}(\omega)\hat{\ell}(\omega)\hat{\ell}(\omega) + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\hat{x}(\omega)\hat{\psi}_{j,q}(\omega)\hat{\psi}_{j,q}(\omega) \\ &= \hat{x}(\omega)\left[|\hat{h}(\omega)|^2 + |\hat{\ell}(\omega)|^2 + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}|\hat{\psi}_{j,q}(\omega)|^2\right] \\ &= \hat{x}(\omega).\end{aligned}$$

Taking inverse Fourier Transform gives result. □

It turns out that the inverse formula (3.4) is based on the adjoint of the wavelet transform. Let us explain. For $x, y \in \mathbf{L}^2(\mathbb{R}^2)$ define their inner product as:

$$\langle x, y\rangle := \int_{\mathbb{R}^2} x(u)\overline{y(u)}\, du, \tag{3.5}$$

where $\overline{x(u)}$ is the complex conjugate of $x(u)$. Using this inner product we can define an inner product on $\mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$. Write $F \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$ as

$$F = \{f_h\,,\ f_J\,,\ f_{j,q} : 0 \le j < J,\ \ 0 \le q < Q\},$$

where $f_h \in \mathbf{L}^2(\mathbb{R}^2)$, $f_J \in \mathbf{L}^2(\mathbb{R}^2)$, and $f_{j,q} \in \mathbf{L}^2(\mathbb{R}^2)$ for each j and q. Then for $F, G \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$, define their inner product as:

$$\langle F, G\rangle := \langle f_h, g_h\rangle + \langle f_J, g_J\rangle + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\langle f_{j,q}, g_{j,q}\rangle,$$

where the inner products on the right hand side are defined using (3.5). Notice that with this inner product we have

$$\|W_{J,Q}x\|^2 = \langle W_{J,Q}x, W_{J,Q}x\rangle.$$

Now let us define the adjoint of the wavelet transform. The adjoint of $W_{J,Q} : \mathbf{L}^2(\mathbb{R}^2) \to \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$ is the map $W^*_{J,Q} : \mathbf{L}^2(\mathbb{R}^2)^{JQ+2} \to \mathbf{L}^2(\mathbb{R}^2)$ such that the following relation holds:

$$\forall\, x \in \mathbf{L}^2(\mathbb{R}^2),\ \ \forall\, F \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}, \quad \langle W_{J,Q}x, F\rangle = \langle x, W^*_{J,Q}F\rangle. \tag{3.6}$$

Using (3.6) as the definition of $W^*_{J,Q}$, one can prove the following theorem.

Theorem 9. The adjoint of $W_{J,Q}$ is

$$W^*_{J,Q}F = f_h * h + f_J * \ell_J + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1} f_{j,q} * \psi_{j,q}, \quad \forall\, F \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}.$$

Proof. Let $x \in \mathbf{L}^2(\mathbb{R}^2)$ and $F \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$. The filters are real symmetric. We can also use

Fubini's Theorem.

$$
\begin{aligned}
&\langle W_{J,Q}x, F\rangle \\
&= \langle (x * h), f_h\rangle + \langle (x * \ell_J), f_\ell\rangle + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\langle (x * \psi_{j,q}), f_{j,q}\rangle \\
&= \int_{\mathbb{R}^2} (x * h)(u)\overline{f_h(u)}\,du + \int_{\mathbb{R}^2} (x * \ell_J)(u)\overline{f_\ell(u)}\,du + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2} (x * \psi_{j,q})(u)\overline{f_{j,q}(u)}\,du \\
&= \int_{\mathbb{R}^2}\left(\int_{\mathbb{R}^2} x(t)h(u-t)\,dt\right)\overline{f_h(u)}\,du + \int_{\mathbb{R}^2}\left(\int_{\mathbb{R}^2} x(t)\ell_J(u-t)\,dt\right)\overline{f_\ell(u)}\,du \\
&\quad + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2}\left(\int_{\mathbb{R}^2} x(t)\psi_{j,q}(u-t)\,dt\right)\overline{f_{j,q}(u)}\,du \\
&= \int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_h(u)}h(u-t)\,du\right)dt + \int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_\ell(u)}\ell_J(u-t)\,du\right)dt \\
&\quad + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_{j,q}(u)}\psi_{j,q}(u-t)\,du\right)dt \\
&= \int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_h(u)}h(t-u)\,du\right)dt + \int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_\ell(u)}\ell_J(t-u)\,du\right)dt \\
&\quad + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2} x(t)\left(\int_{\mathbb{R}^2}\overline{f_{j,q}(u)}\psi_{j,q}(t-u)\,du\right)dt \\
&= \int_{\mathbb{R}^2} x(t)\left(\overline{\int_{\mathbb{R}^2} f_h(u)h(t-u)\,du}\right)dt + \int_{\mathbb{R}^2} x(t)\left(\overline{\int_{\mathbb{R}^2} f_\ell(u)\ell_J(t-u)\,du}\right)dt \\
&\quad + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2} x(t)\left(\overline{\int_{\mathbb{R}^2} f_{j,q}(u)\psi_{j,q}(t-u)\,du}\right)dt \\
&= \int_{\mathbb{R}^2} x(t)\cdot\overline{f_h * h(t)}\,dt + \int_{\mathbb{R}^2} x(t)\cdot\overline{f_\ell * \ell_J(t)}\,dt + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\int_{\mathbb{R}^2} x(t)\cdot\overline{f_{j,q} * \psi_{j,q}(t)}\,dt \\
&= \langle x, (f_h * h)\rangle + \langle x, (f_\ell * \ell_J)\rangle + \sum_{j=0}^{J-1}\sum_{q=0}^{Q-1}\langle x, (f_{j,q} * \psi_{j,q})\rangle \\
&= \langle x, W_{J,Q}^* F\rangle.
\end{aligned}
$$

□

Looking back at (3.4) we see that the left inverse of $W_{J,Q}$ is given by $W_{J,Q}^*$, i.e.,

$$x = W_{J,Q}^* W_{J,Q} x.$$

Having the adjoint of $W_{J,Q}$ and this relationship will be useful when we get to the Heeger-Bergen texture synthesis algorithm.

3.3 Heeger Bergen Texture Synthesis

Using the filter bank from before, we can now state the Heeger-Bergen Texture Synthesis algorithm. We will need an auxillary histogram matching algorithm taken from [10]:

Algorithm 3.1 Histogram Matching

1: Start with an input image u and reference image v both of size $M \times N$.
2: Define $L = MN$ and assume that u and v are unraveled.
3: Determine the permutation τ such that $v_{\tau(1)} \leq v_{\tau(2)} \leq \ldots \leq v_{\tau(L)}$.
4: Determine the permutation σ such that $u_{\sigma(1)} \leq u_{\sigma(2)} \leq \ldots \leq u_{\sigma(L)}$.
5: **for** $i = 1$ **to** L **do**
5: $\quad u_{\sigma(i)} \leftarrow v_{\tau(k)}$
6: **end for**

The general idea behind the Heeger Bergen Texture Synthesis algorithm is to match distributions one wavelet coefficients. One starts with a $M \times N$ white noise, say I_W, with pixels sampled from a standard normal distribution and a reference image I_R. One then calculates $W_{J,Q}I_R$ and $W_{J,Q}I_W$, histogram matches corresponding coefficients, and inverts the transform. The general idea is that matching distributions of wavelet coefficients will turn the white noise into an image that is similar to reference texture. Pseudocode for the algorithm is provided in Algorithm 3.2.

Algorithm 3.2 Heeger Bergen Texture Synthesis Algorithm

1: Start with an white noise $I_W \in \mathbb{R}^{M\times N}$, reference image $I_R \in \mathbb{R}^{M\times N}$, set of scales J, number of rotations Q, and number of iterations T.
2: Calculate $W_{J,Q}I_R \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$.
3: **for** $t = 1$ **to** T **do**
4: $\quad$ Calculate $W_{J,Q}I_W \in \mathbf{L}^2(\mathbb{R}^2)^{JQ+2}$.
5: $\quad$ Update $W_{J,Q}I_W$ by histogram matching each element of $W_{J,Q}I_W$ with the corresponding filter in $W_{J,Q}I_R$ (using each filter in $W_{J,Q}I_R$ as the reference histogram).
6: $\quad$ Update I_W via the formula $I_W = W^*_{J,Q}W_{J,Q}I_W$.
7: $\quad$ Histogram match I_W with I_R using I_R as the reference histogram.
8: **end for**

Unfortunately, Algorithm 3.2 does not yield high quality texture synthesis. One possible reason for this is that it does not use a nonlinear filter bank, so complex features in a texture cannot be

effectively captured.

3.4 One Layer Nonlinear Heeger Bergen Texture Synthesis

To improve the synthesis quality of Algorithm 3.2, we propose a slight modification. Consider the nonlinearity $\mathrm{ReLU}(x) := \max\{0, x\}$ and notice that we have the identities

$$|x| = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x) \tag{3.7}$$

and

$$x = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x). \tag{3.8}$$

Now define the nonlinear filter bank

$$W_{J,Q,\varepsilon}x := \{\mathrm{ReLU}(x * \varepsilon h)\,,\ \mathrm{ReLU}(x * \varepsilon \ell_J)\,,\ \mathrm{ReLU}(x * \varepsilon \psi_{j,q}) : 0 \le j < J,\ \ 0 \le q < Q,\ \ \varepsilon = \pm 1\}. \tag{3.9}$$

In particular, for any Ψ in $W_{J,Q}$, we see that

$$|f * \Psi| = \mathrm{ReLU}(f * \Psi) + \mathrm{ReLU}(f * -\Psi) \tag{3.10}$$

and

$$f * \Psi = \mathrm{ReLU}(f * \Psi) - \mathrm{ReLU}(f * -\Psi). \tag{3.11}$$

Thus, it follows that

$$W_{J,Q}x = W_{J,Q,1}x - W_{J,Q,-1}x, \tag{3.12}$$

which means that

$$x = W^*_{J,Q}(W_{J,Q,1}x - W_{J,Q,-1}x). \tag{3.13}$$

Using this inverse formula, we can create the following algorithm. Results using Algorithm 3.3 are given in Figure 3.9.

Algorithm 3.3 ReLU Heeger Bergen Texture Synthesis Algorithm

1: Start with an white noise $I_W \in \mathbb{R}^{M\times N}$, reference image $I_R \in \mathbb{R}^{M\times N}$, set of scales J, number of rotations Q, and number of iterations T.
2: Calculate $W_{J,Q,\varepsilon} I_R \in \mathbf{L}^2(\mathbb{R}^2)^{2(JQ+2)}$.
3: **for** $t = 1$ **to** T **do**
4: Calculate $W_{J,Q,\varepsilon} I_W \in \mathbf{L}^2(\mathbb{R}^2)^{2(JQ+2)}$.
5: Update $W_{J,Q,\varepsilon} I_W$ by histogram matching each element of $W_{J,Q,\varepsilon} I_W$ with the corresponding filter in $W_{J,Q,\varepsilon} I_R$ (using each filter in $W_{J,Q,\varepsilon} I_R$ as the reference histogram).
6: Update I_W via the formula $I_W = W^*_{J,Q}(W_{J,Q,1} I_W - W_{J,Q,-1} I_W)$.
7: Histogram match I_W with I_R using I_R as the reference histogram.
8: **end for**

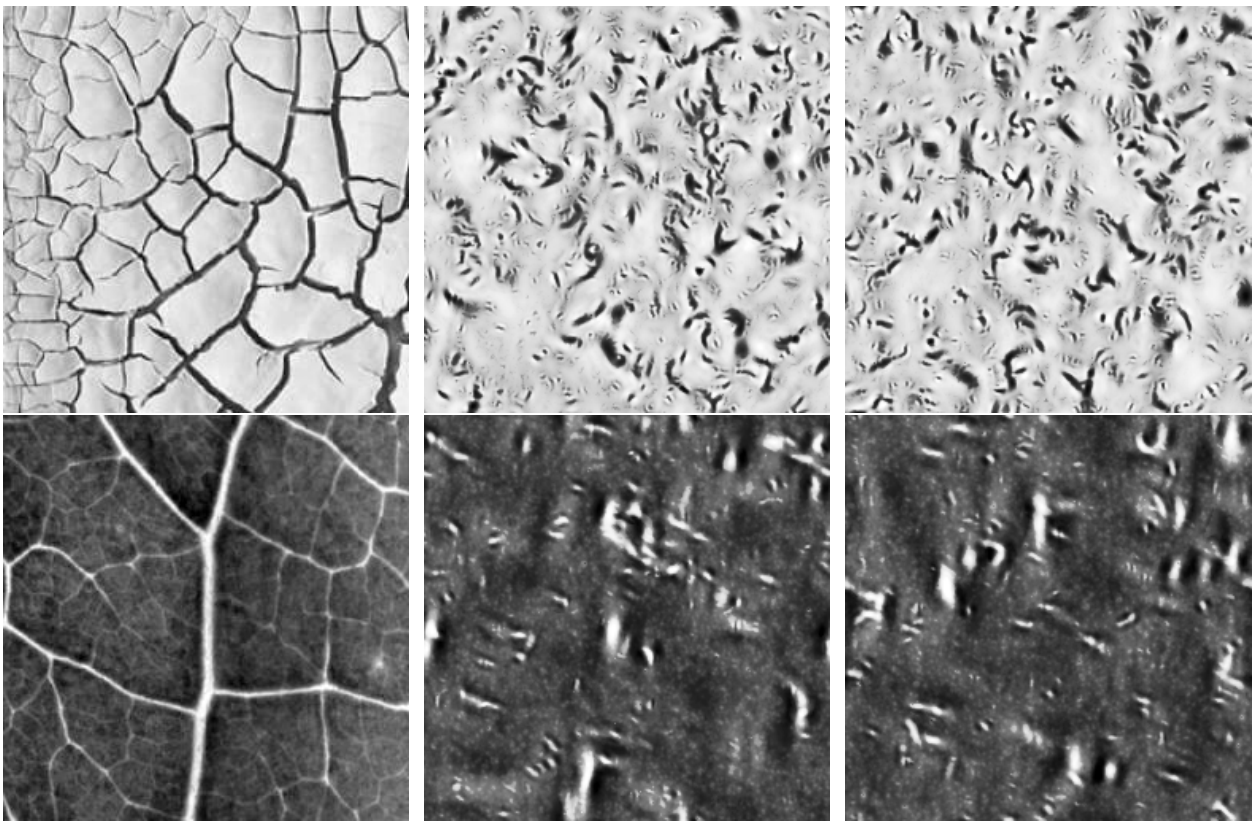


Figure 3.9 **Left:** Reference Texture. **Middle:** No ReLU. **Right:** With ReLU. Ran with $J = 4$ and $Q = 6$ for 50 iterations.

3.5 An Invertible Windowed Scattering Transform

Before we describe the next modification to the Heeger Bergen Texture Synthesis algorithm, we need to motivate the usage of wavelet scattering transforms. We will use the following notation

again:

$$L_c f(x) = f(x - c) \tag{3.14}$$

$$L\tau f(x) = f(x - \tau(x)). \tag{3.15}$$

The first operator, L_c, is a translation operator, and the second operator L_τ can be thought of as a deformation operator. In particular, if

$$\|\nabla\tau\|_\infty = \sup_{x\in\mathbb{R}^n} |\nabla\tau(x)| < 1 \tag{3.16}$$

and $\tau \in \mathbf{C}^2(\mathbb{R}^n)$. Suppose that $\mathcal{H}_1, \mathcal{H}_2$ are a Hilbert spaces and $\Phi : \mathcal{H}_1 \to \mathcal{H}_2$ is an operator for a vision-related task, such as classification. We would like Φ to have the following properties:

- **Local Translation Invariance:** small translations of an object should not greatly affect the output of Φ.
- **Nonexpansiveness:** for $f, g \in \mathcal{H}_1$,

  $$\|\Phi f - \Phi g\|_{\mathcal{H}_2} \leq \|f - g\|_{\mathcal{H}_1}. \tag{3.17}$$

  In other words, the distance between Φf and Φg should not be larger than the original distance between f and g for stability reasons.
- **Stability to Diffeomorphisms:** there exists $C > 0$ such that

  $$\|\Phi f - \Phi L_\tau f\|_{\mathcal{H}_2} \leq C\|f\|_{\mathcal{H}_1}(\|\tau\|_\infty + \|\nabla\tau\|_\infty + \|H\tau\|_\infty). \tag{3.18}$$

  for all $\tau \in \mathbf{C}^2(\mathbb{R}^n)$. That is, the operator "linearizes" small deformations.

As shown in [34], the Fourier modulus is not stable to small dilations, which are a simple class of diffeomorphisms. A natural next step is to use wavelets, which produce a representation that is well localized in space and frequency.

Let G^+ be the set of "positive" rotations and define

$$\Lambda_J = \begin{cases} \{\lambda = 2^j r : r \in G^+, j < J\} \text{ if } J < \infty \\ \{\lambda = 2^j r : r \in G^+, j \in \mathbb{Z}\} \text{ if } J = \infty. \end{cases} \tag{3.19}$$

Let ψ be a wavelet. For $x \in L^2(\mathbb{R}^n)$, let $\lambda \in \Lambda_J$ and define

$$U[\lambda]x = |x * \psi_\lambda| \tag{3.20}$$

A path p is an ordered tuple $p = (\lambda_1, \ldots \lambda_\ell)$ with $\lambda_i \in \Lambda_\infty$ for $i = 1 \ldots, \ell$, and P_J where each element of the path is from Λ_J. For paths, define the operator

$$\begin{aligned} U[p]x &= U[\lambda_\ell] \ldots U[\lambda_2]U[\lambda_1]x \\ &= |||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \ldots | * \psi_{\lambda_\ell}|. \end{aligned} \tag{3.21}$$

Let $\phi_J(u)$ be a low pass filter. For $p \in P_J$, define the windowed scattering operator

$$S_J[p]x(u) = (U[p]x * \phi_J)(u) \tag{3.22}$$

For any set of paths, say Ω, define the path set $S_J[\Omega] = \{S_J[p]\}_{p\in\Omega}$. We now define the windowed scattering transform as the set $S_J[P_J]x$ and use the following Hilbert Space norm

$$\|S_J[\Omega]x\|^2 = \sum_{p\in\Omega} \|S_J[p]x\|_2^2. \tag{3.23}$$

Notably, the windowed scattering transform has many properties we would like in a feature extractor.

Theorem 10 (The Windowed Scattering Norm is Well-defined)**.** Suppose that ψ is a wavelet such that there $\eta \in \mathbb{R}^n$, $\rho \geq 0$, $|\hat{\rho}(\omega)| \leq |\hat{\phi}(2\omega)|$, $\hat{\rho}(0) = 1$, and

$$\hat{\Psi}(\omega) = |\hat{\rho}(\omega - \eta)|^2 - \sum_{k=1}^{\infty} k \left(1 - |\hat{\rho}(2^{-k}\omega - \eta)|^2\right) \tag{3.24}$$

satisfies

$$\alpha - \inf_{1\leq\omega\leq 2} \sum_{j=-\infty}^{\infty} \sum_{r\in G} \hat{\Psi}(2^{-j}r^{-1}\omega)|\hat{\psi}(2^{-j}r^{-1}\omega)|^2 > 0. \tag{3.25}$$

We will call this the admissibility condition. Under the admissibility condition, for all $x \in \mathbb{L}^2(\mathbb{R}^n)$, we have the isometry

$$\|S_J[P_J]x\| = \|x\|. \tag{3.26}$$

Theorem 11 (Nonexpansive Property of the Windowed Scattering Transform)**.** For all $x, y \in L^2(\mathbb{R}^n)$,

$$\|S_J[P_J]x - S_H[P_J]y\| \leq \|x - y\|_2. \tag{3.27}$$

Theorem 12 (Local Translation Invariance of the Windowed Scattering Transform)**.** For all $c \in \mathbb{R}^n$, $x \in L^2(\mathbb{R}^n)$, and an admissible wavelet,

$$\lim_{J\to\infty} \|S_J[P_J]x - S_J[P_J]L_c x\| = 0. \tag{3.28}$$

Theorem 13 (Diffeomorphism Stability)**.** Let $\tau \in C^2(\mathbb{R}^n)$ with $\|D\tau\|_\infty < \frac{1}{2n}$. For an admissible wavelet and any $x \in L^2(\mathbb{R}^n)$, there exists a constant C such that

$$\|S_J[P_J]L_\tau x - S_J[P_J]x\| \leq CK(\tau)\|x\|, \tag{3.29}$$

where $K(\tau) \to 0$ as

$$\|\tau\|_\infty + \|\nabla\tau\|_\infty + \|H\tau\|_\infty \to 0.$$

That is to say, a windowed scattering operator is a good feature extractor. In practice, it is not able to compute an infinite cascade of wavelet transforms, but empirical studies have shown that two layers is enough [33].

We will now construct a two layer modification of the scattering transform, which we will denote as the "Two Layer Scattering Pyramid," using the operators $W_{J,Q}$ and $W_{J,Q,\varepsilon}$:

$$S_2 x := \{x * \phi_J, x * h, W_{J,Q}\mathrm{ReLU}(x * \varepsilon\psi_{j,q}) \,:\, 0 \leq j \leq J-1\,,\, 0 \leq q \leq Q-1\,,\, \varepsilon = \pm 1\}. \tag{3.30}$$

The algorithm is provided in Algorithm 3.4. Note that this can be generalized to multiple layers, but the computational cost is infeasible; additionally, most of energy should be in the first two layers.

Algorithm 3.4 Two Layer Scattering Pyramid

INPUTS: An image x and operators $W_{J,Q}$, $W_{J,Q,-1}$, $W_{J,Q,1}$.
OUTPUT: S_2x, a modification of the Two Layer Scattering Pyramid.
Initialize a set of functions S_2x.
Calculate $W_{J,Q,-1}x$ and $W_{J,Q,1}x$ and add

$$x * \phi_J = \text{ReLU}(x * \phi_J) - \text{ReLU}(x * -\phi_J)$$

and

$$x * h = \text{ReLU}(x * h) - \text{ReLU}(x * -h).$$

for $j = 0$ **to** $J - 1$ **do**
  for $q = 0$ **to** $Q - 1$ **do**
    for $\varepsilon = \pm 1$ **do**
      Calculate $W_{J,Q,\epsilon}(x * \psi_{j,q})$ and add to S_2x.
    end for
  end for
end for
Return S_2x.

Algorithm 3.5 Two Layer Scattering Pyramid Inverse

INPUTS: The two layer pyramid S_2x.
OUTPUT: The original image x.
for $j = 0$ **to** $J - 1$ **do**
  for $q = 0$ **to** $Q - 1$ **do**
    Calculate $W^*_{J,Q}(W_{J,Q,1}(x * \psi_{j,q}) - W_{J,Q,-1}(x * \psi_{j,q}))$, where corresponding filters are subtracted in the subtraction operation.
  end for
end for
Note that this recovers all the bandpass filters in $W_{J,Q}x$, and the high and low pass residuals are already in S_2x.
$x = W^*_{J,Q}W_{J,Q}x$.
Return x.

For the next section, we will use the algorithms above to formulate a modified texture synthesis algorithm.

3.6 Two Layer Nonlinear Heeger Bergen Texture Synthesis

Using Algorithm 3.4, we can formulate a Heeger Bergen Texture Synthesis algorithm motivated by the scattering transform.

Algorithm 3.6 Heeger Bergen Scattering Texture Synthesis

Start with an white noise $I_W \in \mathbb{R}^{M \times N}$, reference image $I_R \in \mathbb{R}^{M \times N}$, set of scales J, number of rotations Q, and number of iterations T.
Calculate $S_2 I_R$ and save it.
for $t = 1$ **to** T **do**
    Calculate $W_{J,Q,1} I_W$ and $W_{J,Q,-1} I_W$.

    Update $W_{J,Q,1} I_W$ by histogram matching each element of $W_{J,Q,1} I_W$ with the corresponding filter in $W_{J,Q,1} I_R$ (using each filter in $W_{J,Q,1} I_R$ as the reference histogram) and update $W_{J,Q,-1} I_W$ by histogram matching each element of $W_{J,Q,-1} I_W$ with the corresponding filter in $W_{J,Q,-1} I_R$ (using each filter in $W_{J,Q,-1} I_R$ as the reference histogram).
    Keep

$$I_W * \phi_J = \text{ReLU}(I_W * \phi_J) - \text{ReLU}(I_W * -\phi_J)$$

    and

$$I_W * h = \text{ReLU}(I_W * h) - \text{ReLU}(I_W * -h)$$

    for inversion.
    for $j = 0$ **to** $J - 1$ **do**
        for $q = 0$ **to** $Q - 1$ **do**
            for $\varepsilon = \pm 1$ **do**
                Calculate $W_{J,Q,\epsilon}(I_W * \varepsilon \psi_{j,q})$ and histogram match with corresponding filter in $S_2 I_R$.
            end for
        end for
    end for
    Invert the scattering pyramid $S_2 I_W$ via 3.5.
end for
Histogram match I_W with I_R using I_R as the reference histogram.

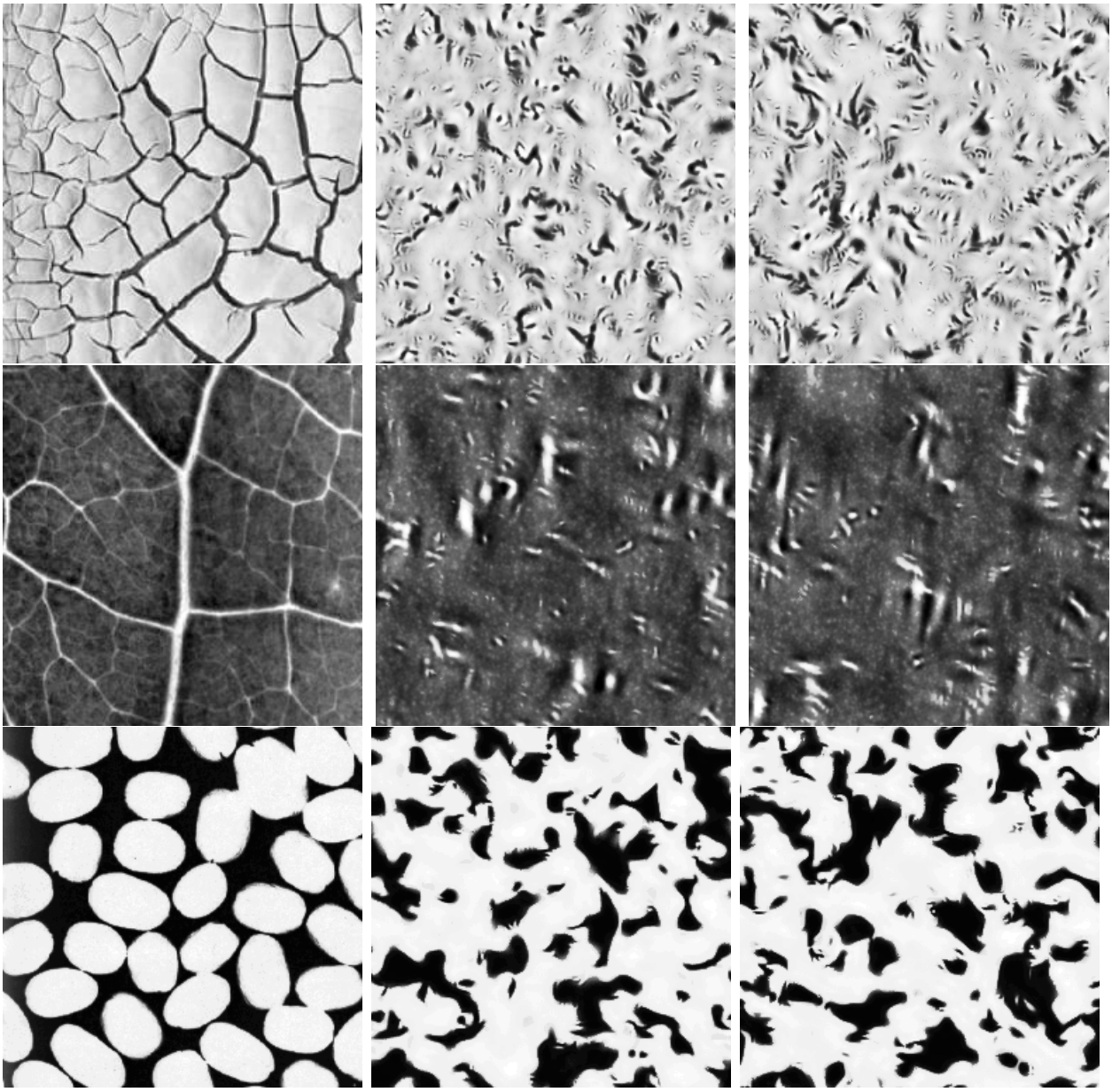


Figure 3.10 **Left:** Reference Texture. **Middle:** One Layer ReLU Synthesis for 50 iterations. **Right:** Two Layer Synthesis for 10 iterations. Ran with $J = 4$ and $Q = 6$.

3.7 Conclusions and Future Work

Based on our current experiments, we believe there are multiple reasons why our approach fails. First, matching wavelet coefficients is an approximation of optimal transport. However, this approximation seems to get stuck in local minimums quickly. There are two possible reasons

- Histogram matching wavelet coefficients is bad approximation of optimal transport.
- Matching each coefficient works, but they "interfere" each other. That is, matches in one coefficiens might lead to better synthesis, but a match in a different coefficient leads to worse synthesis.

The second point leads to an interesting question. Two representations can have close to the same histogram for each component, but their structure does not necessarily look the same; what else do we need to match to obtain structure such as edges and lines?

One approach, which we will consider for deep neural networks in the next chapter, is to match n-dimensional histograms. We can regard the set of wavelet coefficients as a $H \times W \times C$ tensor with each $H \times W$ slice of the last dimension being a wavelet coefficient. If we unroll the vector into a $HW \times C$ matrix and match HW dimension histograms for each pixel, it is a good approximation of an optimal transport problem. Will this problem yield good results for synthesis? Our preliminary experiments suggest yes. If this is the case, can we study what type of frames yield good texture synthesis? Is invertibility/pseudoinvertibility really a necessary requirement?

CHAPTER 4

LONG RANGE CONSTRAINTS FOR NEURAL TEXTURE SYNTHESIS USING SLICED WASSERSTEIN LOSS

This work is based on [51], which has been submitted to IEEE ICIP 2023. We introduce an algorithm for texture synthesis using a modified form of Sliced Wasserstein Loss. The main idea of both methods is similar: we match 1D histograms between feature maps. However, our approach for this chapter will use deep convolutional neural networks (CNNs) instead of an invertible wavelet frame.

4.1 Background on Convolutional Neural Networks

The contents of this section is based on [21]. A CNN is neural network using series of convolutions to learn features from data. The general steps of the network for an image classification task are the following:

- Initialize random weights for a set of filters and put in an input image x.
- Apply a cascade of convolutional layers and nonlinearities using the filters.
- Subsample the result using a subsampling operation such as max pooling, min pooling, or average pooling.
- Repeat this process multiple times.
- Use a classifier (usually feedforward neural network) to classify the image.
- Apply backpropogation to update the filters via gradient descent.
- Repeat the steps till one reaches desired performance on the desired task.

An example of a convolutional architecture, LeNet, is given in Figure 4.1.

First, consider the discrete convolution of functions $x(t)$ and $w(t)$, which are only defined for integer t:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \tag{4.1}$$

Convolutions are done over more than one dimension at a time. In particular, for two dimensions,

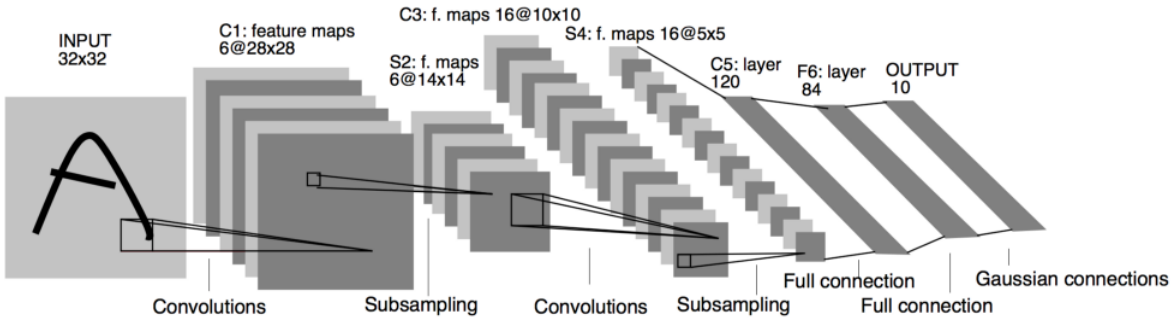


Figure 4.1 LeNet Architecture demonstrating the procedure for the forward pass of a CNN.

for an image two-dimensional image I and two dimensional kernel K, the convolution is

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n), \tag{4.2}$$

which is a commutative operation.

However, neural network libraries usually use the cross-correlation function

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n). \tag{4.3}$$

due to ease of implementation. While the operation is not commutative, this does not affect the accuracy of a network because the kernel K is learned from the data anyway. In practice, when ones uses a convolutional neural network, we regard an image or feature map as an 3-tensor. For example, an $M \times N$ RGB image is an $M \times N \times 3$ tensor, where the last dimension is known as the channel dimension. Each channel of a the three tensor (i.e. each $M \times N$ matrix) is known as a feature map.

We can now provide a mathematical formulation for a convolutional layer of a network. Assume we have an $M \times N \times C_1$ input. In other words, there are C_1 feature maps in the channel dimension. If we would like to output C_2 feature maps from our convolutional layer, define the set of filters $\{F_{i,j}\}$ with $1 \leq i \leq C_1$ and $1 \leq j \leq C_2$. A the "convolution" step of convolutional layer is

$$C(x) = \sum_{i=1}^{C_1} F_{i,j} * x, \tag{4.4}$$

which has channel dimension C_2. After the application of the convolution step above, one applies

a pointwise nonlinearity, σ to get

$$\sigma(C(x)) = \sigma\left(\sum_{i=1}^{C_1} F_{i,j} * x\right). \tag{4.5}$$

Common choices are the sigmoid function, tanh, ReLU, and so on.

After a convolutional layer, CNNs usually have a subsampling operation to decrease the size of the feature maps. This makes training networks less computationally intensive and helps with extracting features such as edges. Usually, a pooling operation, such as max pooling, which involves sliding a kernel and taking the maximum along a window, is used.

The reason CNNs work better than, such as feedfoward neural networks, for vision tasks is that they encode information in a more relevant manner. Specifically, the convolution step of a convolutional layer is translation equivariant (e.g. applying a translation to an image before and after the convolution step yields the same result); convolutions are a local operation, and local information is more relevant for images; a feature map shares parameters with other feature maps, which captures interaction between different feature maps and lowers the total number of learned parameters. Parameter sharing and more capacity for larger networks are some of the reasons one would suspect a deep CNN to work better than a scattering transform for classification tasks.

Regarding deep CNNs, we are interested in a specific architecture which is commonly used for neural texture synthesis tasks, VGG19 [45]. See Figures 4.2 and 4.3.

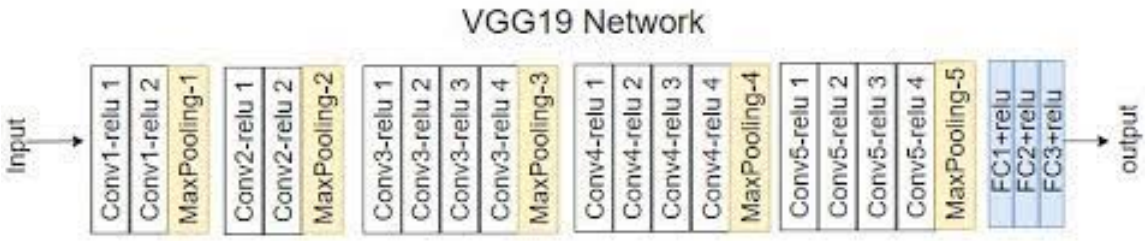


Figure 4.2 Schematic of VGG19 architecture. The blocks "Conv" are a convolution block and ReLU, "MaxPooling" is a max pooling operation, and "FC+relu" is a linear layer and ReLU.

4.2 Image Quality Metrics

To evaluate the effectiveness of our texture synthesis algorithms, discuss a few image quality metrics that we will use to compare the quality of images we generated. Traditional image metrics,

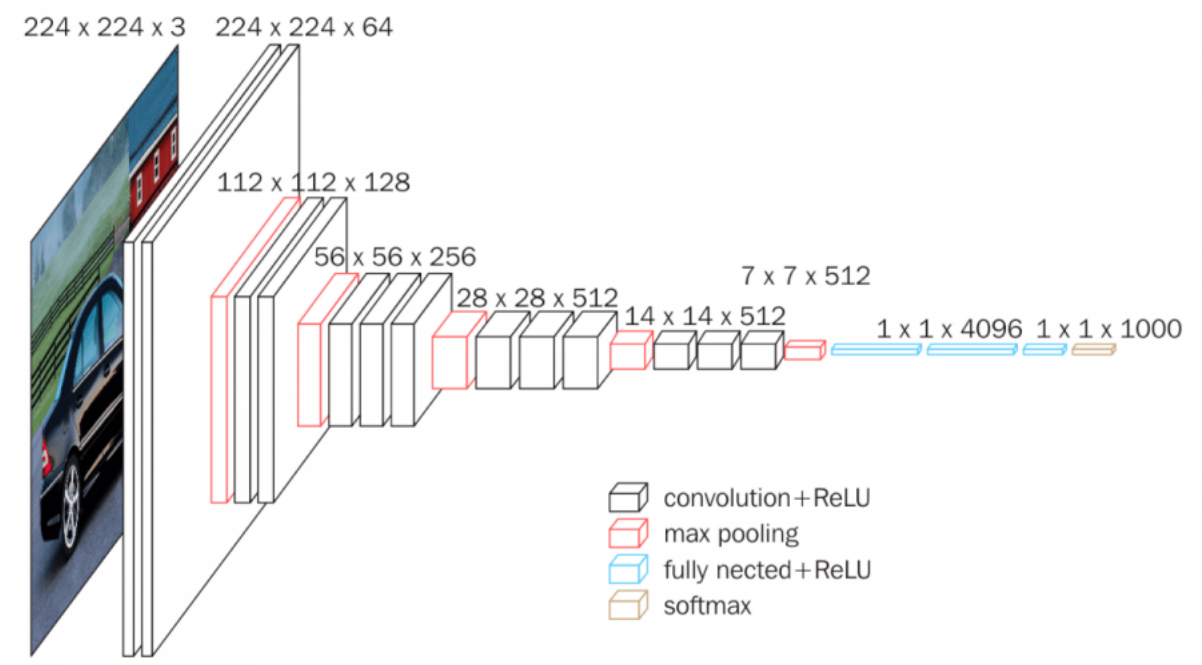


Figure 4.3 VGG19 schematic showing how feature map size decreases as the number of filters increase.

like PSNR and SSIM, are not perfect image metrics. Consider the following example:

Figure 4.4 Two examples where human perception does not match with common image metrics.

In the example, one can clearly see that Patch 1 looks similar to the reference, and is slightly deformed. However, common image metrics like MSE would not be able to handle the distortions properly. Additionally, Patch 0, which is a low pass filtering of the reference, would be recognized as similar by traditional metrics. Note that these metrics we will discuss are not optimal either. In all cases, if we calculate the similarity between two of the same image, the image metric would

yield the best score. Like we mentioned previously, this is not ideal for texture synthesis.

The first metric we consider is LPIPS [52], which measures the perceptual similarity between two images. The idea is to use MSE between feature maps of a deep convolutional neural network.

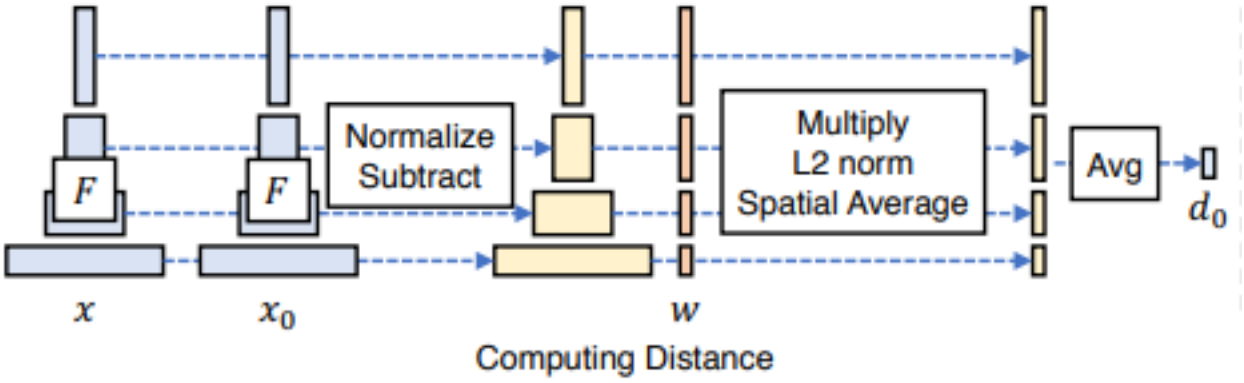


Figure 4.5 LPIPS calculation between an image x and x_0.

Suppose we have $\ell = 1, \ldots, L$ layers of a neural network. We extract feature stacks from layer ℓ and unit-normalize in the channel dimension, which we designate $y^\ell, \hat{y}^\ell \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$. We do a channel-wise scaling of the activiations by vector $w^\ell \in \mathbb{R}^{C_\ell}$ Finally, we average spatially and over the channels. This can be represented as

$$d(x, x_0) = \sum_{\ell=1}^{L} \frac{1}{H_\ell W_\ell} \sum_{i=1}^{H_\ell} \sum_{j=1}^{W_\ell} \|w_\ell \circ (y^\ell - \hat{y}^\ell)\|_2^2 \tag{4.6}$$

Usually, a deep network like VGG or AlexNet are used for feature map extraction.

The second metric we use is Frechet Inception distance (FID) [25]. The 2-Wasserstein Distance, or Frechet Distance, is given by

$$d_F^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|^2 d\gamma(x, y) \tag{4.7}$$

where $\Gamma(\mu, \nu)$ is the set of joint probability measures such that

$$\left\{ \gamma : \int_{\mathbb{R}^n} \gamma(x, y) = dy = \mu(x) \text{ and } \mathbb{R}^n \} \gamma(x, y) = dx = \nu(y) \right\}.$$

For FID, one assumes that the model fits to a Gaussian distribution. Then for two Gaussian distributions p_1 with distribution $\mathcal{N}(\mu_1, \Sigma_1)$ and p_2 with distribution $\mathcal{N}(\mu_2, \Sigma_2)$, one can write the

FID as

$$d_F^2(p_1, p_2) = \|\mu_1 - \mu_2\|_2^2 + \text{tr}\left[\Sigma_1 + \Sigma_1' - 2\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\right]. \tag{4.8}$$

Usually, the function to estimate the distribution is a specific layer of a deep convolutional neural network, similar to LPIPS.

KID score [8] is similar to FID score, but relaxes the assumption that both distributions are Gaussian and is defined as

$$\text{KID}(p_1, p_2) = \text{MMD}(p_1, p_2)^2. \tag{4.9}$$

More detail is given in [8], but it will be omitted from this thesis.

4.3 Texture Synthesis Using VGG19 and Gram Matrices

The contents of this section are adapted from [19], which has made led to remarkable improvements in the field of texture synthesis since its publication. For notation, assume we have an $L + 1$ layer convolutional neural network; for $\ell = 0, \ldots, L$, layer ℓ has N_ℓ distinct feature maps. Each of these feature maps is a M_ℓ vector after flattening. We reshape feature maps in layer ℓ into a matrix $F_\ell \in \mathbb{R}^{N_\ell \times M_\ell}$, where each row of the matrix is a feature map.

Assuming that each row of F_ℓ (i.e. each feature map in layer ℓ) is mean centered, we define the Gram matrix $G_\ell \in \mathbb{R}^n \times \mathbb{R}^n$ as

$$G_{ij}^\ell = \sum_k F_{ik}^\ell F_{jk}^\ell. \tag{4.10}$$

In matrix form, we have $G^\ell = F^\ell (F^\ell)^T$.

To generate a new texture from a reference texture, we update a white noise image using gradient descent by optimizing over the mean-squared error (MSE) between the Gram matrices of the reference image and the white noise. More formally, let $\vec{x}$ be a reference image and $\hat{\vec{x}}$ be the generated image. Also let the gram matrices for $\vec{x}$ be G^ℓ and the gram matrices for $\hat{\vec{x}}$ be $\hat{g}^\ell$. Define

$$E_\ell = \frac{1}{4N_\ell^2 M_\ell^2} \sum_{i,j} (G_{ij}^\ell - \hat{G}_{ij}^\ell)^2, \tag{4.11}$$

which is the MSE between the gram matrices of $\vec{x}$ and $\hat{\vec{x}}$ in layer ℓ. The full loss function is given

by

$$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{\ell=0}^{L} w_\ell E_\ell, \tag{4.12}$$

where w_ℓ are user-defined weights.

For updating the image, one uses gradient descent. In particular, the following identity holds:

$$\frac{\partial E_\ell}{\partial \hat{F}_{ij}^\ell} = \begin{cases} \frac{1}{N_\ell^2 M_\ell^2} \left((\hat{F}^\ell)^T \left(G^\ell - \hat{G}^\ell \right)_{ji} \right), & \hat{F}_{ij}^\ell > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

In practice, one does not implement the gradient manually. Instead, one can use open-source deep learning packages, like PyTorch or Tensorflow, with a built in optimizer (L-BFGS) to update the white noise image.

Before we provide a unified workflow, we discuss the specific CNN architecture, which is a modified version of VGG-19, used for texture synthesis. The following changes were made to the original version of VGG-19:

- The max pooling layers are switched to average pooling layers. No retraining of the network is done. The authors noticed that this change improved image quality.
- The weights of the network are transformed so that the mean of each feature map is 1.

Additionally, during the synthesis process the feature maps from the layers 'conv1_1', 'pool1', 'pool2', 'pool3', and 'pool4' are used to make Gram matrices. The other convolutional layers and the fully connected part of the network are not used.A schematic of the general workflow is given in Figure 4.6 and some synthesis results are given in Figure 4.7.

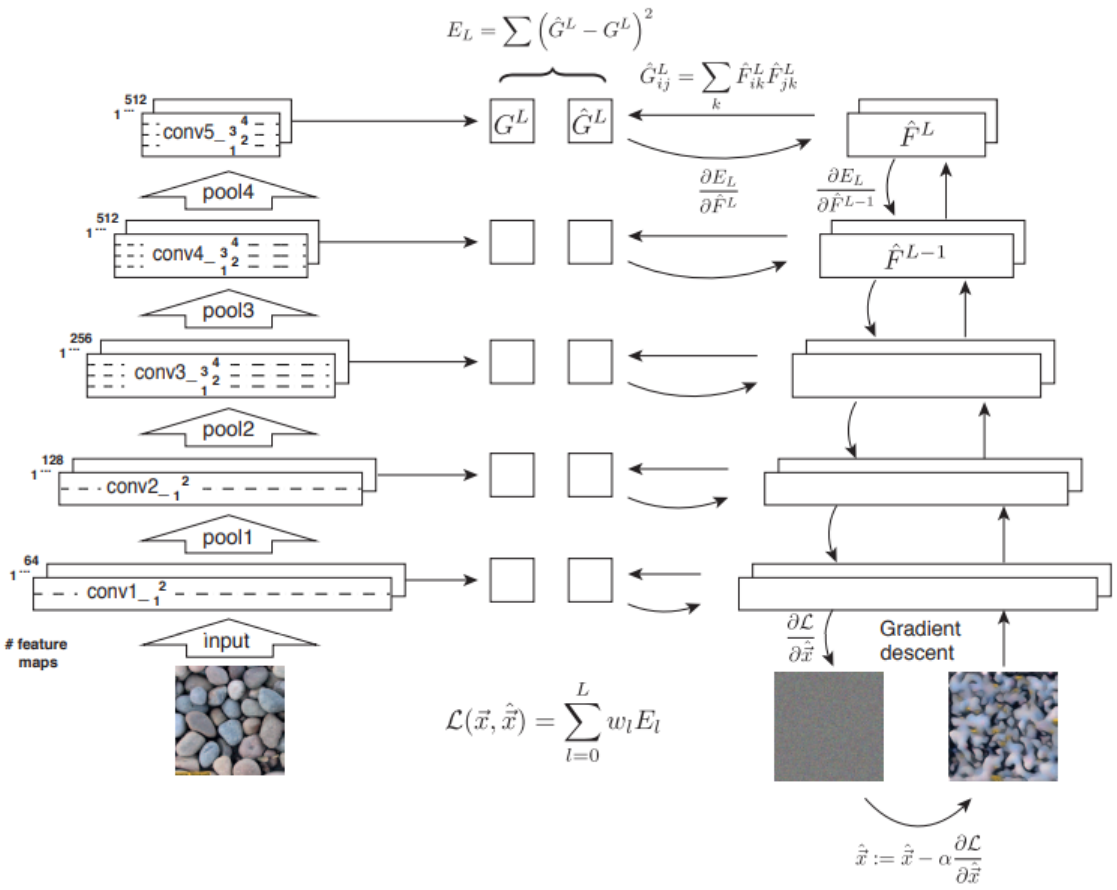


Figure 4.6 The general synthesis method is given below. The images $\vec{x}$ and $\hat{\vec{x}}$ are both passed through the network. Gram matrices are created from specific feature maps, and the loss is calculated. Using a backpropogation engine, the $\hat{\vec{x}}$ is updated. Picture reference: [19].

Figure 4.7 Results for textures synthesis after each gram matrix is added for synthesis. "Original" denotes the reference image and "Portilla and Simoncelli" are results from [40]. Picture reference: [19].

One notable issue with the results above is that long range constraints are not captured by this algorithm. In other words, the alignment of objects isn't capture by this algorithm. This is expected because CNNs aggregate local information.

4.4 Improvements Via Regularization

To ameliorate the lack of long range information captured by [19], [31] propose adding regularization (4.12). For an image I, let

$$\mathcal{E}_I = \{\tilde{I} : |\mathcal{F}(I)| = |\mathcal{F}(\tilde{I})|\}. \tag{4.14}$$

That is, (4.14) is the set of images with the same power spectrum as I. We can find the projection of an image $\hat{I}$ onto $\mathcal{E}_I$ via

$$\tilde{I} := \mathcal{F}\left(\frac{\mathcal{F}(\hat{I}) \cdot \mathcal{F}(\hat{I})^*}{|\mathcal{F}(\hat{I}) \cdot \mathcal{F}(\hat{I})^*|} \cdot \mathcal{F}(I)\right). \tag{4.15}$$

Let $d(\hat{I}, \mathcal{E}_I)$ be the distance of $\hat{I}$ to $\mathcal{E}_I$. If we denote $\mathcal{L}_{\text{CNN}}$ to be the loss from (4.12), then our new loss function between images is

$$\mathcal{L} = \mathcal{L}_{\text{CNN}} + \frac{\beta}{2} d(\hat{I}, \mathcal{E}_I)^2, \tag{4.16}$$

where β is a hyperparameter.

In Figure 4.8, some examples of synthesis using the spectrum constraint are given. As shown in Figure 4.8, there is notable improvement in alignment for these textures. However, it is difficult to find a value for β that works on a variety of images. This would require time-consuming hyperparameter tuning. Additionally, if one required high quality synthesis for multiple textures, one may have to tune β depending on each texture. It would be ideal if one could find a method that did not require hyperparameter tuning for high quality synthesis results.

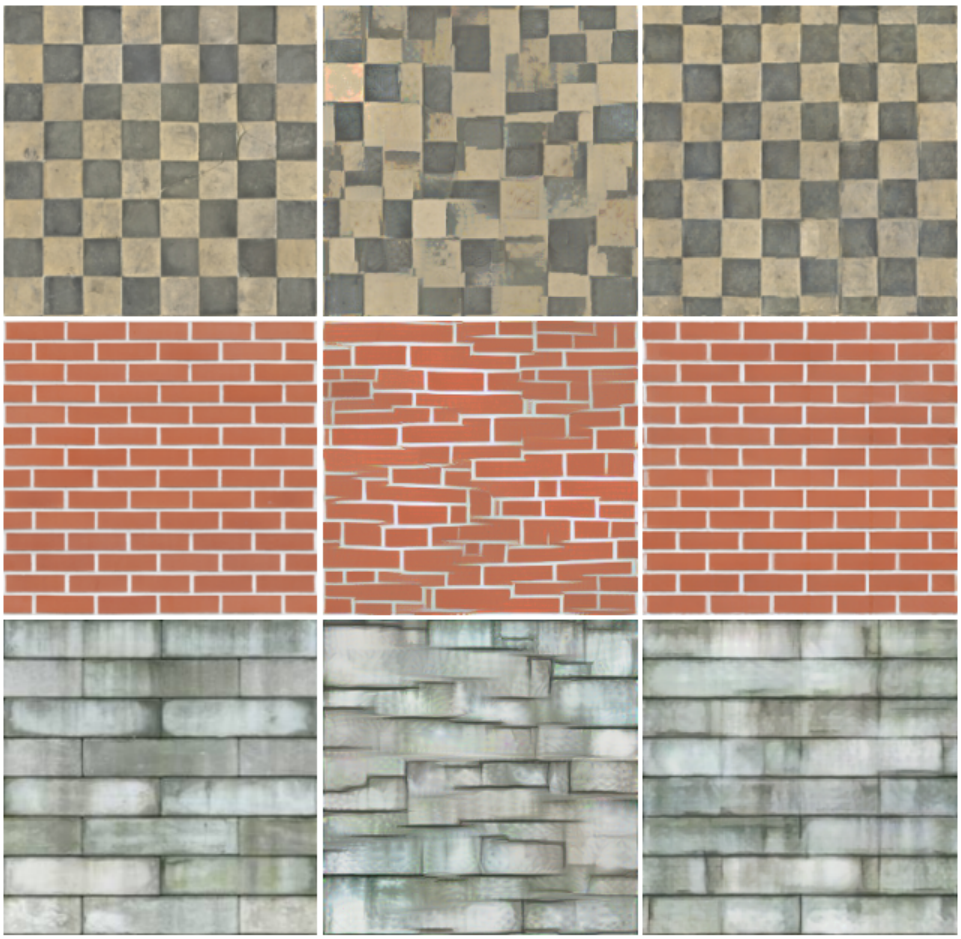


Figure 4.8 Results for textures synthesis using a spectrum constraint. The hyperparameter β is 10^5. **Left:** Reference Texture. **Middle:** Without Spectrum Constraint. **Right:** With Spectrum Constraint.

4.5 Texture Synthesis Using Sliced Wasserstein Loss

Before we discuss the method from [51], we will provide some background information about texture synthesis using Sliced Wasserstein Loss from [24]. Suppose that layer ℓ of an L layer convolutional neural network has N_ℓ channels and M_ℓ pixels in each channel. We denote the feature vector located at pixel m as $F_m^\ell \in \mathbb{R}^{N_\ell}$, which is a change in notation from previous sections.

The authors of [24] propose using a different set of statistics instead of the mean squared error between gram matrices. In a manner similar to [23, 48], the authors match the distributions between feature maps, but these feature maps used in [24] are feature maps of VGG19 [45].

With respect a network architecture, let p^ℓ and $\hat{p}^\ell$ be the probability density functions associated

with the set of feature vectors $\{F_m^\ell\}$ and $\{\hat{F}_m^\ell\}$. Since our network is discrete, we assume that the probability density functions are always an average of Dirac delta distributions of the form

$$p^\ell(x) = \frac{1}{M_\ell} \sum_{m=1}^{M_\ell} \delta_{F_m^\ell}(x). \tag{4.17}$$

Let $V \in \mathbb{S}^{N_\ell}$ be a random direction on the unit sphere of dimension N_ℓ. For the purpose of this paper, the Sliced Wasserstein Distance between two distributions of features is of the form

$$\mathcal{L}_{\mathrm{SW},\ell}(p^\ell, \hat{p}^\ell) = \mathbb{E}_V[\mathcal{L}_{\mathrm{SW1D}}(p_V^\ell, \hat{p}_V^\ell)], \tag{4.18}$$

where

$$p_V^\ell := \{\langle F_m^\ell, V\rangle\} \tag{4.19}$$

is a set consisting of batched projections of the feature maps F_m^ℓ onto the directions V; if we make a vector P_V^ℓ consisting of the elements of p_V^ℓ, the $1D$ Sliced Wasserstein Loss is the 2-norm between sets of sorted projections:

$$\mathcal{L}_{\mathrm{SW1D}}(p_V^\ell, \hat{p}_V^\ell) = \frac{1}{\mathrm{len}(P_V^\ell)} \left\|\mathrm{sort}(P_V^\ell) - \mathrm{sort}(\hat{P}_V^\ell)\right\|_2^2 \tag{4.20}$$

and the full Sliced Wasserstein loss over all the layers is

$$\mathcal{L}_{\mathrm{SW}}(I_1, I_2) = \sum_{\ell=1}^{L} \mathcal{L}_{\mathrm{SW},\ell}(p_{V,I_1}^\ell, p_{V,I_2}^\ell), \tag{4.21}$$

for images I_1 and I_2, respectively. For practical applications, one uses a loss of the form

$$\mathcal{L}_{\mathrm{SW}}(I_1, I_2) = \sum_{\ell=1}^{L} w_i \mathcal{L}_{\mathrm{SW},\ell}(p_{V,I_1}^\ell, p_{V,I_2}^\ell), \tag{4.22}$$

where w_i are weight terms that set to zero for layers that are not used. We will use this formulation for the rest of the paper.

An implementation of 4.20 is provided below in pseudocode:

Algorithm 4.1 Implementation of 4.20

Set I_{WN} as variable to be updated by optimizer.
for $k = 1, \ldots, M$ **do**
    Calculate Extract(I_{WN}).
    Calculate Extract(I_{ref}).
    Calculate $\mathcal{L}_{\text{Slicing}}(I_{\text{WN}}, I_{\text{ref}})$.
    Backpropogate and update I_{WN}.
end for
Return updated I_{WN} as synthesized texture.

Pitie et al. [39] showed that Sliced Wasserstein Distance satisfies:

$$\mathcal{L}_{\text{SW}}(p, \hat{p}) = 0 \implies p = \hat{p}.$$

The same does not hold for other losses used for texture synthesis, such as the gram matrix loss. Thus, using a Sliced Wasserstein-based loss should capture more stationary statistics compared to the traditional Gram Loss.

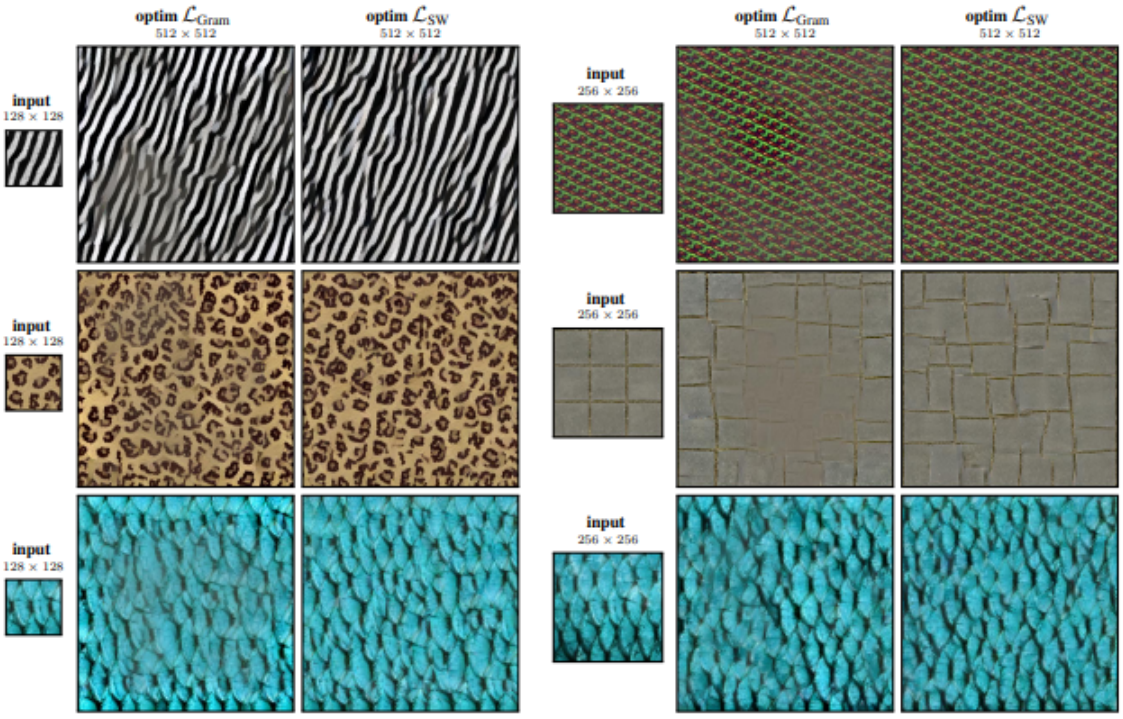


Figure 4.9 Synthesis results using the gram matrix loss, denoted $\mathcal{L}_{\text{Gram}}$ and using $\mathcal{L}_{\text{SW}}$.

In Figure 4.9, some examples of synthesis using sliced wasserstein loss are given above. While the alignment is better for the synthesis, the algorithm still has trouble capturing long range

constraints, like alignment. The authors propose adding a spatial tag as fourth channel in an RGB image to guide synthesis, which is shown in Figure 4.10.

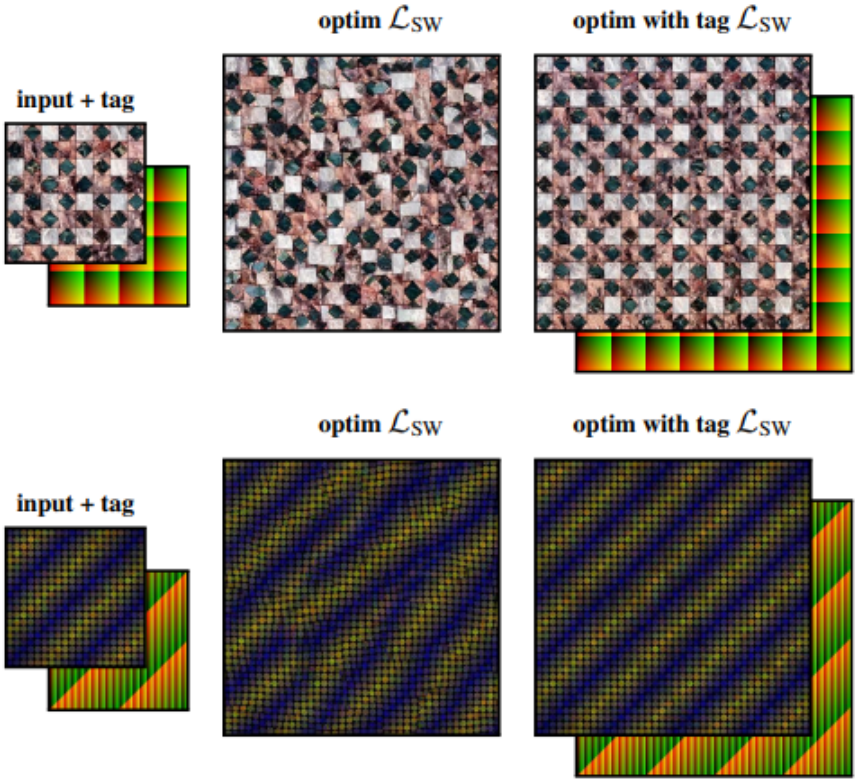


Figure 4.10 Synthesis results with and without a spatial tag.

While adding a spatial tag creates visually appealing textures, one needs a spatial tag for each constraint they wish to impose. However, it's unlikely to have a prepared spatial tag for each texture one would like to generate, especially if the texture is highly irregular. Thus, it would be ideal if there was an algorithm that could capture long range constraints without user guidance and without tedious hyperparameter tuning.

4.6 New Texture Synthesis Algorithm

The method proposed in [24] cannot effectively capture long range constraints unless a user-added spatial tag is added to guide synthesis. This is most likely Sliced Wasserstein Loss does not fully capture nonstationary statistics in an image. Thus we propose a new set of statistics for texture synthesis based on Sliced Wasserstein Loss to capture long range constraints without any

supervision or hyperparameter tuning. Our experiments show that our proposed set of statistics provides competitive results with current approaches. Additionally, we augment our synthesis results via a coarse-to-fine multi-scale procedure, which yield state-of-the-art results.

Instead of just matching distributions via slicing over the channel dimension of the feature maps, we purpose matching more statistics in a very simple way. Consider a set of feature maps $F^\ell \in \mathbb{R}^{H_\ell \times W_\ell \times N_\ell}$. In the original algorithm, we unravel each $H_\ell \times W_\ell$ feature map, consider each pixel m to get feature vectors of length N_ℓ, and project onto direction V in Eq. (4.21). Another way to reshape the feature maps is to unravel them into H_ℓ different $W_\ell \times N_\ell$ feature maps, F^ℓ_H (with $F^\ell_{H,n}$ being a vector of all n^{th} pixels of each feature vector), and project them onto $V_{H_\ell} \in \mathbb{S}^{H_\ell}$. Analogous to Eq. (4.19), for the distribution p^ℓ_H associated to feature vectors $\{F^\ell_{H,n}\}$ we can define another set of batched projections given by

$$p^\ell_{V_{H_\ell}} = \{\langle F^\ell_{H,n}, V_{H_\ell}\rangle\}. \tag{4.23}$$

The corresponding additional loss term is

$$\mathcal{L}_{\text{SW},H}(I_1, I_2) = \sum_{\ell=1}^{L} w_i \mathcal{L}_{\text{SW},\ell}\left(p^\ell_{V_{H_\ell},I_1}, p^\ell_{V_{H_\ell},I_2}\right). \tag{4.24}$$

Intuitively, this loss term accounts for alignment in an image by slicing over the dimension for the height of the feature maps rather than the dimension for the channel of the feature maps. The new loss function we consider is

$$\mathcal{L}_{\text{Slicing}}(I_1, I_2) = \mathcal{L}_{\text{SW}}(I_1, I_2) + \mathcal{L}_{\text{SW},H}(I_1, I_2), \tag{4.25}$$

which is the sum of Eq. (4.21) and Eq. (4.24).

Denote the feature map extraction from VGG19 as Extract(I). For our algorithm, we start with a reference image I_{ref} and a white noise I_{WN} and run for M epochs. Our implementation for slicing synthesis is the same as in [24] for Eq. (4.21) where we perform a batch projection on N_ℓ directions and sort. For our additional loss term in equation Eq. (4.24), the number of batched projections we make is H_ℓ. In our next algorithm, assume that the goal is to synthesize an image the same size as the reference image without any loss of generality.

Algorithm 4.2 Synthesis Algorithm

1: Set I_{WN} as variable to be updated by optimizer.
2: **for** $k = 1, \ldots, M$ **do**
3: Calculate Extract(I_{WN}).
4: Calculate Extract(I_{ref}).
5: Calculate $\mathcal{L}_{\text{Slicing}}(I_{\text{WN}}, I_{\text{ref}})$.
6: Backpropogate and update I_{WN}.
7: **end for**
8: Return updated I_{WN} as synthesized texture.

The settings for the slicing loss are to use the first 12 layers of VGG19 for calculating $\mathcal{L}_{\text{SW}}$ and the first two convolutions (after the ReLU) in each convolution block for for calculating $\mathcal{L}_{\text{SW},H}$. The L-BFGS optimizer [30] is used for optimization with a learning rate of $\eta = 1$.

We start by comparing results with [24]. We use the author's TensorFlow implementation, which is a previous commit in https://github.com/tchambon/A-Sliced-Wasserstein-Loss-for-Neural-Texture-Synthesis. without their spatial tag on some relatively periodic textures. For all the experiments in this paper, our texture sources were the following:

- https://github.com/omrysendik/DCor/tree/master/Data
- https://www.robots.ox.ac.uk/ vgg/data/dtd

for generating 256×256 textures. The results are given in Fig. 4.11.

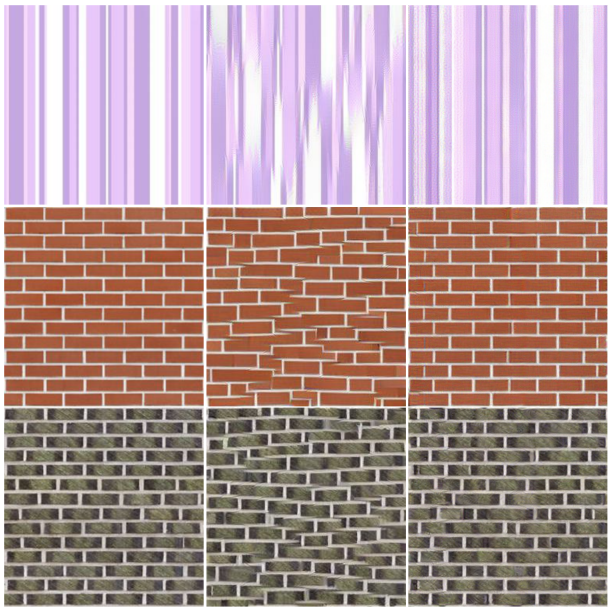


Figure 4.11 Original SW loss vs. Eq. (4.25). **Left:** Reference. **Mid:** SW Loss. **Right:** Eq. (4.25) Loss (Ours).

However, there are still some textures that are not generated perfectly. See Fig. 4.12 for some examples.

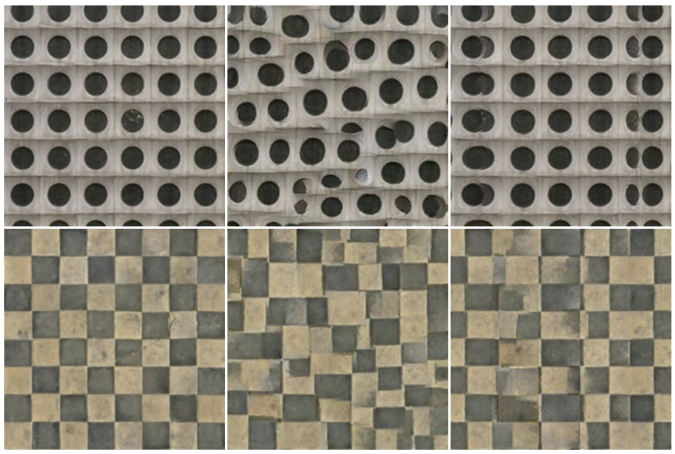


Figure 4.12 Less successful cases. **Left:** Reference. **Mid:** SW Loss. **Right:** Eq. (4.25) Loss (Ours).

Now the results of using Eq. (4.25) is compared to [31]. Note that we use the implementation from [20] for this comparison. as an additional point of comparison. There is no comparison with [43] because experiments from [43, 20] have shown the algorithm does not yield much

improvement for nonperiodic textures. Take note of the top row of Figure 4.13 in particular. The

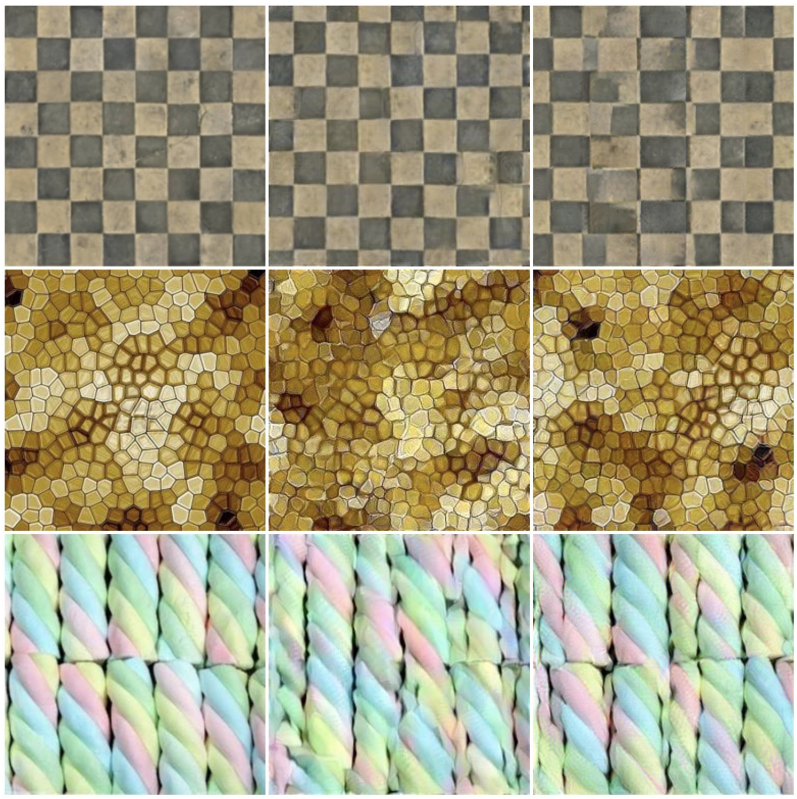


Figure 4.13 Gram Matrix + Spectrum vs. Eq. (4.25) Loss. **Left:** Reference. **Mid:** Spectrum Constraint. **Right:** Eq. (4.25) Loss (Ours).

spectrum constraint produces better results, which suggests that there could be improvement in our proposed synthesis method.

Quantitative comparisons between the original SW loss, the spectrum constraint, and our proposed method using a set of 34 images are provided in this section. The LPIPS [52], FID [25], crop-based FID. This is done by taking sixty-four 128×128 crops of the reference texture and synthesized texture for each exemplar (the FID/KID score is calculated between these two sets of images. For the ground truth case, a different set of crops of the reference is used) like [32], KID [8], and crop-based KID (c-KID) score are provided in Table 4.1. For FID and KID based scores, the implementation from [37] is used. In Table 4.1, SW stands for the method using the original SW Loss, Spec. stands for using a spectrum constraint, and GT stands for the Ground Truth.

Table 4.1 Quantitative Comparison

Method	LPIPS	FID	c-FID	KID	c-KID
Ours	**0.437**	107.220	**71.938**	−0.014	**0.073**
SW	0.454	101.768	78.683	**−0.016**	0.083
Spec.	0.447	**99.615**	78.250	**−0.016**	0.083
GT	0	0	18.069	−0.025	0

From the table, our results are competitive and our proposed set of statistics *did not require searching for a proper hyperparameter to get competitive results*. Note that our results for FID and c-FID vary compared to [32] because FID is a biased estimate [14] and our sample count is very low.

4.7 Improvements via a Multi-scale Approach

Since there is room for improvement in our synthesis, we augment our algorithm with a multi-scale procedure in a manner identical to [20, 49]. For the multi-scale algorithm at K scales, let $I_{\text{ref},i}$ be the reference image downsampled by a scale factor of 2^i with $i = 0, \ldots, K$, and define the upsampling operator as $\text{Upsample}(I)$. Lastly, define the output of Algorithm 4.2 using the notation $I_{\text{Synthesis}} = \text{SWSynthesis}(I_{\text{input}}, I_{\text{ref}})$, where I_{input} is the input to be optimized via backpropogation, I_{ref} is the reference texture, and $I_{\text{Synthesis}}$ is the output after synthesis using Algorithm 1. The results are given in Figure 4.14.

Algorithm 4.3 Multi-scale Synthesis Algorithm

1: Initialize $I_{\text{Synthesis}}$ as a white noise that is the same size as the reference texture downsampled by 2^K.
2: **for** $i = 0, \ldots, K$ **do**
3: $\quad I_{\text{Synthesis}} \leftarrow \text{SWSynthesis}(I_{\text{Synthesis}}, I_{\text{ref},K-i})$.
4: $\quad I_{\text{Synthesis}} \leftarrow \text{Upsample}(I_{\text{Synthesis}})$.
5: **end for**
6: Return $I_{\text{Synthesis}}$ as the synthesized texture.

In Figure 4.14, note the small improvements in edge generation and general structure when using $K = 1$ compared to $K = 0$.

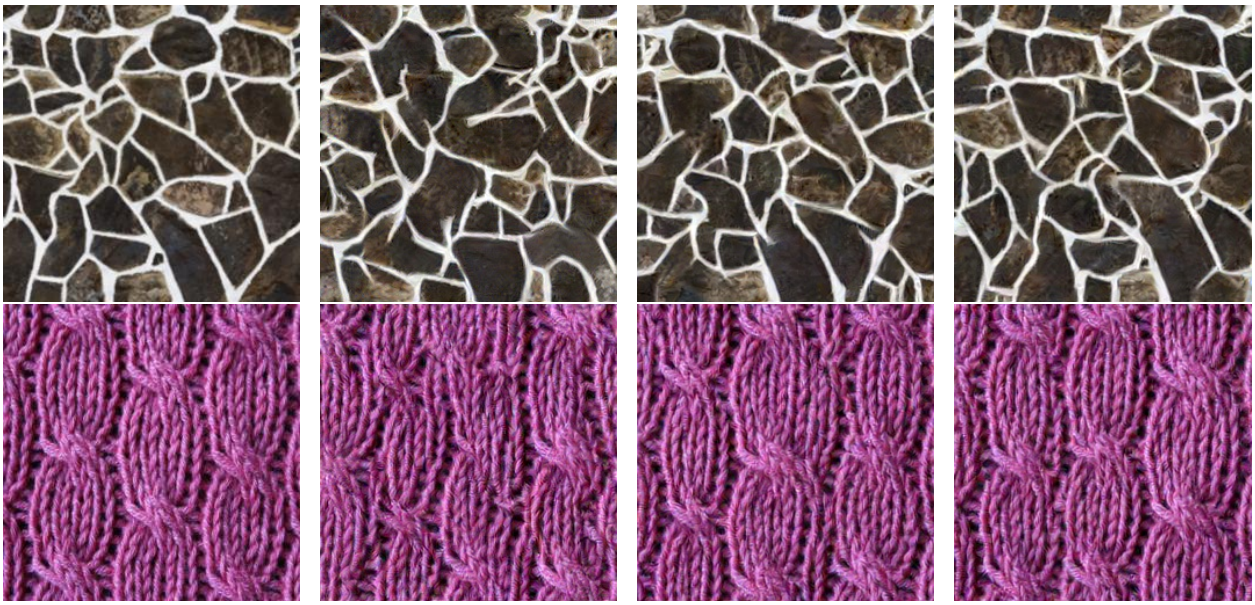


Figure 4.14 Multi-scale procedure at different scales. **Left:** Reference. **Mid Left:** $K = 0$. **Mid Right:** $K = 1$. **Right:** $K = 2$.

The intuition for why it works is that it generates the details of the texture in a coarse-to-fine way; the initial scale generates the general color and macro-scale features and additional scales add on fine-grain details in an image.

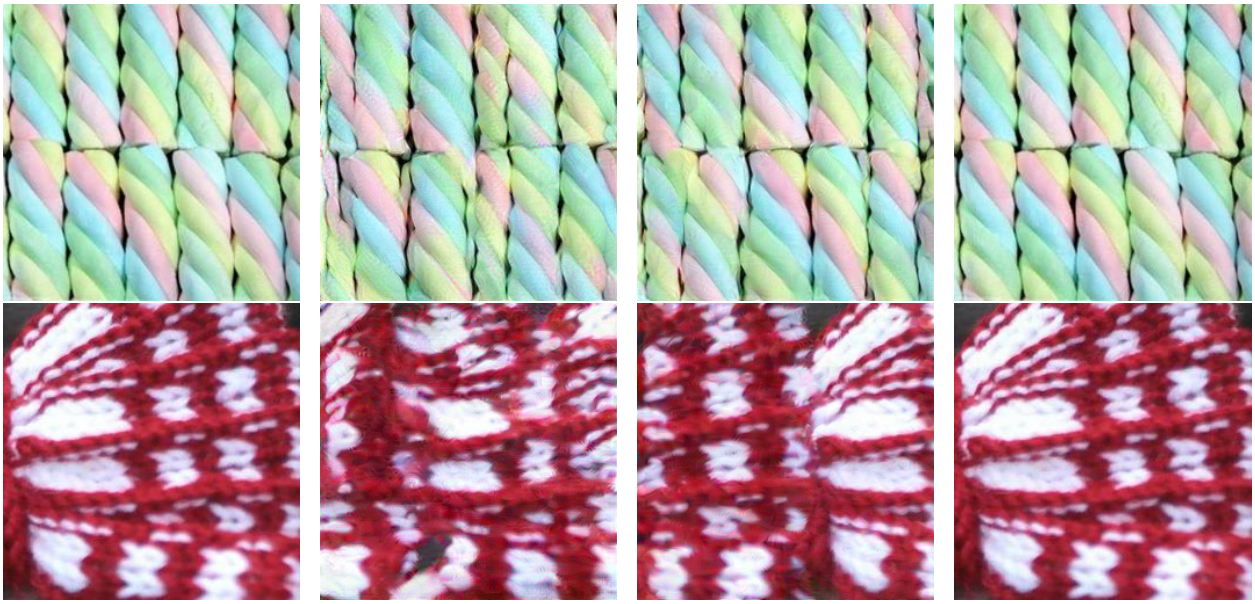


Figure 4.15 Progression of synthesis that lead to repetitions. **Left:** Reference Texture. **Middle Left:** $K = 0$. **Middle Right:** $K = 1$. **Right:** $K = 2$.

However, it is possible to create replica textures for larger values of K. Of the 34 images

generated for the experiments, there were four repetitions when $K = 2$. See Figure 4.15 for examples.

Additionally, a quantitative comparison using the same image quality metrics from before is provided. Unlike in previous comparisons, it would be better to have a higher LPIPS score at a comparable generative metric; this would mean that our textures are less likely to be replicas, but still have similar qualities. The results are given in Table 4.2.

Table 4.2 Quantitative Score for $K = 0, 1, 2$

Scale	LPIPS	FID	c-FID	KID	c-KID
$K = 0$	0.437	107.220	71.938	−0.014	0.073
$K = 1$	0.381	67.118	53.908	−0.018	0.044
$K = 2$	0.250	38.304	40.220	−0.022	0.027

Based on the scores and Fig. 5, $K = 1$ provides a nice mix between diversity and image quality at our fixed image size.

In the Figure 4.16, the multi-scale approach is applied without the additional loss term in Eq. (4.25).

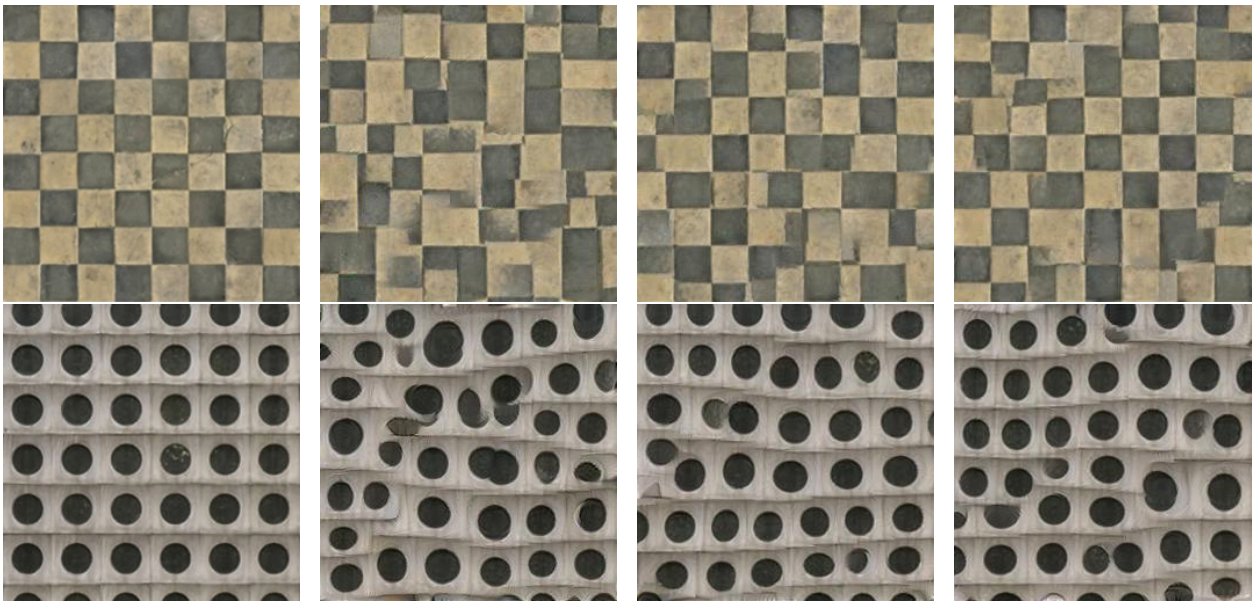


Figure 4.16 Results with SW Loss. **Left:** Reference. **Mid Left:** $K = 0$. **Mid Right:** $K = 1$. **Right:** $K = 2$.

From our results, the multi-scale approach by itself is not enough to fully capture nonstationary statistics or enforce long range constraints. That is to say, the loss term added in Eq. (4.25) is absolutely necessary to capture long-range constraints in textures.

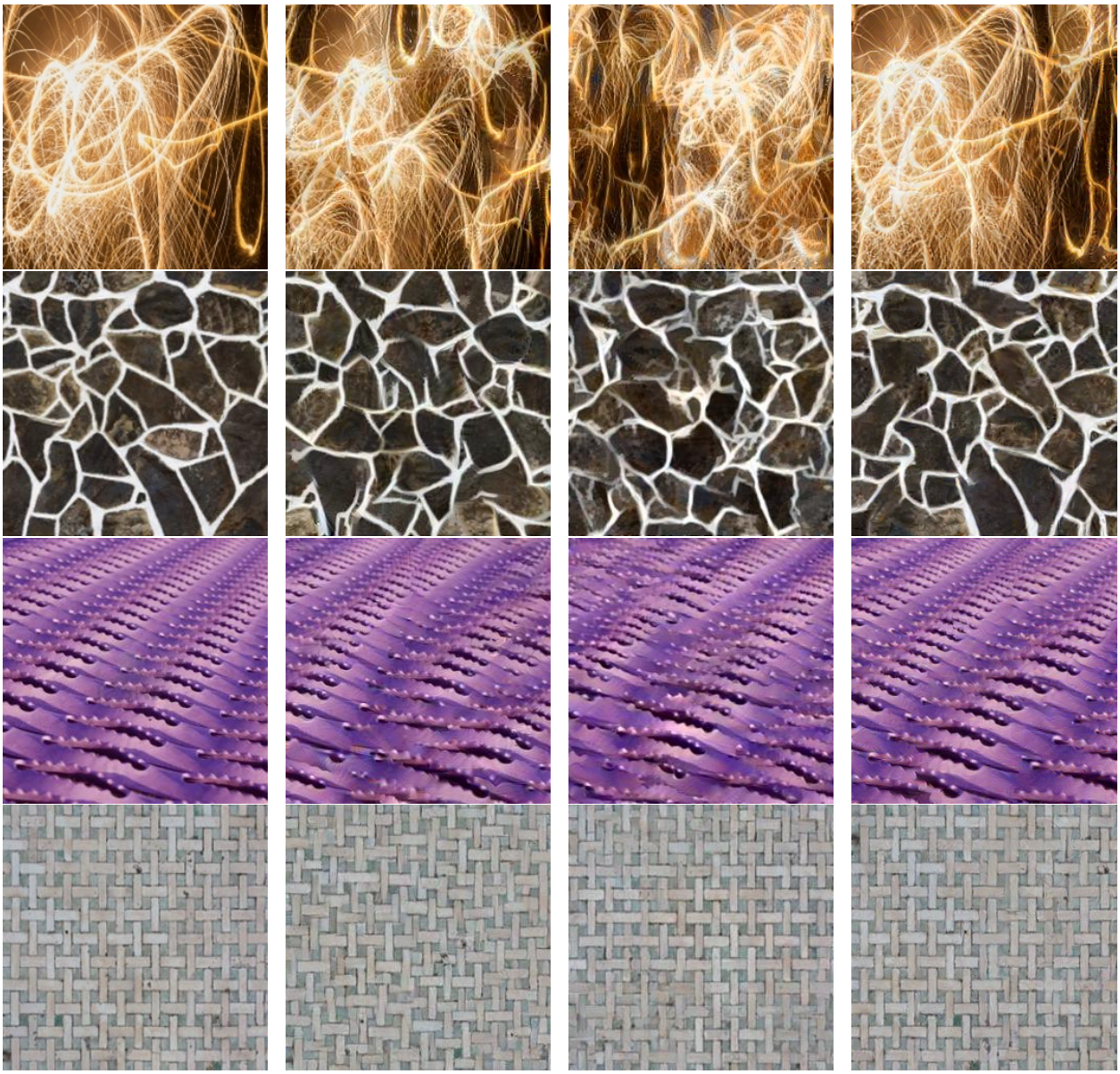


Figure 4.17 Comparison of results. **Left:** Reference. **Mid Left:** SW Loss. **Mid Right:** Gonthier. **Right:** $K = 1$ (Ours).

Lastly, the results using $K = 1$ are compared with [20] using the default settings from their experiments ($K = 2$) to show the effectiveness our mutli-scale results relative to another multi-scale algorithm. The results from [24] again for an additional point of reference. Some results are shown

in Figure 4.17 and a quantitative study is given in Table 4.3.

Table 4.3 Comparison of $K = 1$, SW Loss, Gonthier

Method	LPIPS	FID	c-FID	KID	c-KID
$K = 1$	**0.381**	**67.118**	**53.908**	**−0.018**	**0.044**
SW	0.454	101.768	78.683	−0.016	0.083
Gonthier	0.415	77.569	67.728	**−0.018**	0.067
GT	0	0	18.069	−0.025	0

4.8 Conclusions

We present a modification of texture synthesis via Sliced Wasserstein Loss that has the ability to add long range constraints without user-added spatial tags (supervision). Our additional loss term can be thought of as a regularization term, but unlike traditional regularization terms, one does not need to hyperparameter tune to enforce long range constraints. That is to say, the proposed method requires less user supervision for competitive results. One thing we have not tested is whether the number of scales is dependent on image size. We believe this is true, and it is probable that one would need to choose the number of scales based on the size of the image.

CHAPTER 5

CONCLUDING REMARKS

We have addressed two important problems in statistical signal processing: mtuli-reference alignment and texture synthesis. For multi-reference alignment, like we mentioned in chapter 2, many open problems remain. In one dimension, one question to consider is how we can approach general diffeomorphisms. While considering a specific set of group actions seems to be the most viable approach, one wonders whether it would be possible to consider limiting the size of $\|\tau'\|_\infty$ and $\|\tau''\|_\infty$, which would make τ "close" to a translation. Additionally, is it possible to generalize all our results to two and three dimensions? Both these cases would be more relevant to practitioners in cryo-EM.

Regarding texture synthesis, one wonders whether a deep representation is actually needed. That is, could we use a wavelet transform or scattering transform with sliced wasserstein loss to generate textures? Our preliminary experiments, which are not available in this thesis, suggest this is possible. However, the synthesis quality is not as strong. This suggests that we could use a different filter bank for better synthesis. VGG models have omnidirectional filters and all the filters are not necessarily positive, which we believe is responsible for strong synthesis. Future work will focus on testing our hypotheses.

BIBLIOGRAPHY

[1] Emmanuel Abbe, Tamir Bendory, William Leeb, João M Pereira, Nir Sharon, and Amit Singer. Multireference alignment is easier with an aperiodic translation distribution. *IEEE Transactions on Information Theory*, 65(6):3565–3584, 2018.

[2] Afonso Bandeira, Yutong Chen, Roy R Lederman, and Amit Singer. Non-unique games over compact groups and orientation estimation in cryo-em. *Inverse Problems*, 2020.

[3] Afonso S Bandeira, Ben Blum-Smith, Joe Kileel, Amelia Perry, Jonathan Weed, and Alexander S Wein. Estimation under group actions: recovering orbits from invariants. *arXiv preprint arXiv:1712.10163*, 2017.

[4] Afonso S Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Conference on learning theory*, pages 361–382, 2016.

[5] Afonso S Bandeira, Moses Charikar, Amit Singer, and Andy Zhu. Multireference alignment using semidefinite programming. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 459–470. ACM, 2014.

[6] Tamir Bendory, Alberto Bartesaghi, and Amit Singer. Single-particle cryo-electron microscopy: Mathematical theory, computational challenges, and opportunities. *IEEE Signal Processing Magazine*, pages 58–76, March 2020.

[7] Tamir Bendory, Nicolas Boumal, Chao Ma, Zhizhen Zhao, and Amit Singer. Bispectrum inversion with application to multireference alignment. *IEEE Transactions on Signal Processing*, 66(4):1037–1050, 2017.

[8] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*.

[9] Nicolas Boumal. Nonconvex phase synchronization. *SIAM Journal on Optimization*, 26(4):2355–2377, 2016.

[10] Thibaud Briand, Jonathan Vacher, Bruno Galerne, and Julien Rabin. The heeger & bergen pyramid based texture synthesis algorithm. *Image processing on line*, 4:276–299, 2014.

[11] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[12] Yuxin Chen and Emmanuel J Candès. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *Communications on Pure and Applied Mathematics*, 71(8):1648–1714, 2018.

[13] Yuxin Chen, Leonidas J Guibas, and Qi-Xing Huang. Near-optimal joint object matching via convex relaxation. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 100–108, 2014.

[14] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079, 2020.

[15] WB Collis, PR White, and JK Hammond. Higher-order spectra: the bispectrum and trispectrum. *Mechanical systems and signal processing*, 12(3):375–394, 1998.

[16] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[17] Robert Diamond. On the multiple simultaneous superposition of molecular structures by rigid body transformations. *Protein Science*, 1(10):1279–1287, 1992.

[18] Hassan Foroosh, Josiane B Zerubia, and Marc Berthod. Extension of phase correlation to subpixel registration. *IEEE transactions on image processing*, 11(3):188–200, 2002.

[19] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

[20] Nicolas Gonthier, Yann Gousseau, and Saïd Ladjal. High-resolution neural texture synthesis with long-range constraints. *Journal of Mathematical Imaging and Vision*, 64(5):478–492, 2022.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[22] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054, 1982.

[23] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, 1995.

[24] Eric Heitz, Kenneth Vanhoey, Thomas Chambon, and Laurent Belcour. A sliced wasserstein loss for neural texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9412–9420, 2021.

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[26] Matthew Hirn and Anna Little. Unbiasing procedures for scale-invariant multi-reference alignment. *arXiv preprint arXiv:2107.01274*, 2021.

[27] Matthew Hirn and Anna Little. Wavelet invariants for statistically robust multi-reference alignment. *Information and Inference: A Journal of the IMA*, 10(4):1287–1351, 2021.

[28] Zvi Kam. The reconstruction of structure from electron micrographs of randomly oriented particles. In *Electron Microscopy at Molecular Dimensions*, pages 270–277. Springer, 1980.

[29] Fima C Klebaner. *Introduction to stochastic calculus with applications*. World Scientific Publishing Company, 2012.

[30] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[31] Gang Liu, Yann Gousseau, and Gui-Song Xia. Texture synthesis through convolutional neural networks and spectrum constraints. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3234–3239. IEEE, 2016.

[32] Guilin Liu, Rohan Taori, Ting-Chun Wang, Zhiding Yu, Shiqiu Liu, Fitsum A Reda, Karan Sapra, Andrew Tao, and Bryan Catanzaro. Transposer: Universal texture synthesis using feature maps as transposed convolution filter. *arXiv preprint arXiv:2007.07243*, 2020.

[33] Stéphane Mallat. Recursive interferometric representations. In *18th European Signal Processing Conference (EUSIPCO-2010)*, Aalborg, Denmark, 2010.

[34] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, October 2012.

[35] Wooram Park and Gregory S Chirikjian. An assembly automation approach to alignment of noncircular projections in electron microscopy. *IEEE Transactions on Automation Science and Engineering*, 11(3):668–679, 2014.

[36] Wooram Park, Charles R Midgett, Dean R Madden, and Gregory S Chirikjian. A stochastic kinematic model of class averaging in single-particle electron microscopy. *The International journal of robotics research*, 30(6):730–754, 2011.

[37] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

[38] Amelia Perry, Alexander S Wein, Afonso S Bandeira, and Ankur Moitra. Message-passing algorithms for synchronization problems over compact groups. *Communications on Pure and Applied Mathematics*, 71(11):2275–2322, 2018.

[39] F. Pitie, A.C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439 Vol. 2, 2005.

[40] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40:49–70, 2000.

[41] Brian M Sadler and Georgios B Giannakis. Shift-and rotation-invariant object reconstruction using the bispectrum. *JOSA A*, 9(1):57–69, 1992.

[42] Sjors HW Scheres, Mikel Valle, Rafael Nuñez, Carlos OS Sorzano, Roberto Marabini, Gabor T Herman, and Jose-Maria Carazo. Maximum-likelihood multi-reference refinement for electron microscopy images. *Journal of molecular biology*, 348(1):139–149, 2005.

[43] Omry Sendik and Daniel Cohen-Or. Deep correlations for texture synthesis. *ACM Transactions on Graphics (ToG)*, 36(5):1–15, 2017.

[44] Nir Sharon, Joe Kileel, Yuehaw Khoo, Boris Landa, and Amit Singer. Method of moments for 3-D single particle ab initio modeling with non-uniform distribution of viewing angles. *Inverse Problems*, 36(4):044003, 2020.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] Amit Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011.

[47] Amit Singer. Mathematics for cryo-electron microscopy. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3995–4014. World Scientific, 2018.

[48] Guillaume Tartavel, Yann Gousseau, and Gabriel Peyré. Variational texture synthesis with sparsity and spectrum constraints. *Journal of Mathematical Imaging and Vision*, 52:124–144, 2015.

[49] Guillaume Tartavel, Gabriel Peyré, and Yann Gousseau. Wasserstein loss for image synthesis and restoration. *SIAM Journal on Imaging Sciences*, 9(4):1726–1755, 2016.

[50] Douglas L Theobald and Phillip A Steindel. Optimal simultaneous superpositioning of multiple structures with missing data. *Bioinformatics*, 28(15):1972–1979, 2012.

[51] Liping Yin and Albert Chua. Long range constraints for neural texture synthesis using sliced wasserstein loss. *arXiv preprint arXiv:2211.11137*, 2022.

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pages 586–595, 2018.

[53] Yiqiao Zhong and Nicolas Boumal. Near-optimal bounds for phase synchronization. *SIAM Journal on Optimization*, 28(2):989–1016, 2018.

[54] J Portegies Zwart, René van der Heiden, Sjoerd Gelsema, and Frans Groen. Fast translation invariant classification of hrr range profiles in a zero phase representation. *IEE Proceedings-Radar, Sonar and Navigation*, 150(6):411–418, 2003.