ABSTRACT

Title of Dissertation: GENERALIZED SCATTERING TRANSFORMS: STATIONARITY, INVERSION, AND APPLICATIONS

Brandon Christopher Kolstoe
Doctor of Philosophy, 2026

Dissertation Directed by: Professor Wojciech Czaja
Department of Mathematics

This dissertation investigates theoretical and computational aspects of scattering transforms, as well as applications to the problem of hyperspectral reconstruction. Scattering transforms are operators from harmonic analysis with a similar structure to convolutional neural networks but with predefined convolutional filters with known mathematical structure.

The first focus of this dissertation is on generalizing theoretical results, which have previously been proved for scattering transforms with directional wavelet-based convolutional filters, to scattering transforms with more general kinds of filters. First, we prove results on the limiting translation invariance of sequences of scattering transforms with general families of filters. Then, we extend the theory of scattering transforms of strictly stationary random processes from wavelet-based filters to more general filters. In particular, we study stationary scattering transforms whose filters extract time-frequency features of strictly stationary processes, called stationary Fourier scattering transforms.

The second focus of this dissertation is to applications of scattering transforms to the hyper-

spectral reconstruction problem, in which hyperspectral images are generated from smaller and easier-to-obtain images. We develop a pipeline in which wavelet scattering transform features are mapped between these two modalities by a shallow convolutional neural network, followed by a different network which inverts the scattering embedding. We optimize our pipeline hyperparameters over many experiments, and we report our results on the Hyper-Skin and ARAD datasets.

The final focus of this dissertation is dedicated to theoretical and computational aspects of inverting Fourier scattering transforms. For the former, we construct generalized Fourier scattering transforms which are invertible in some cases, and we prove energy decay and bi-Lipschitz results for these transforms. For the latter, we study two different attempts at numerically inverting such transforms.

GENERALIZED SCATTERING TRANSFORMS: STATIONARITY, INVERSION, AND APPLICATIONS

by

Brandon Christopher Kolstoe

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2026

Advisory Committee:
Professor Wojciech Czaja, Chair/Advisor
Professor Radu Balan
Professor Maria Cameron
Professor Vince Lyzinski
Professor Furong Huang, Dean's Representative

Dedication

To my parents and my sparents, without whom I would have never made it this far.

And to my Grandma (and writing partner) Katy. I will always love you.

Acknowledgments

As I reflect on my time in graduate school, I am filled with immense gratitude for everyone who has made this one of the best periods of my life.

First and foremost, I deeply thank my advisor, Professor Wojciech Czaja. His deep insights, wonderful conversations, and unwavering support have been invaluable during my graduate school journey. His strong conviction that mathematics is ultimately done for people has greatly deepened my appreciation and love of both applied and pure mathematics, and his guidance has been essential in the construction of this dissertation.

I sincerely thank the members of my committee, Professor Radu Balan, Professor Maria Cameron, Professor Vince Lyzinski, and Professor Furong Huang, for spending their time reviewing this thesis. In particular, I thank Professor Radu Balan, with whom I have had many helpful conversations and who also helped introduce me to harmonic analysis, the true queen of mathematics.

Thank you also to Dr. Richard G. Spencer, from the National Institute on Aging at the NIH, for his years of funding support and fruitful collaborations. Several ideas in this dissertation arose from our conversations, and I deeply valued his patience as I learned how to present mathematics to non-mathematicians.

I have had the immense fortune of being able to work with many fellow students and postdocs of my advisor, including Jeremiah Emidih, Canran Ji, Fushuai Jiang, David Koralov, Randy Martinez, Sanghoon Na, Shashank Sule, Gabriel Vilarrubi, Noah Waldman, and Matthias Wellershoff, who have provided me with wonderful collaborations and amazing discussions. I have also had many other friends without whom my graduate school experience would have been far dimmer, including (but certainly not limited to) Max Auer, Gonzalo Benavides, Baidehi Chattopadhyay, Oscar Coppola, Ben Goldschlager, Khalil Guy, Revati Jadhav, Emily Jiang, Frank McBride, Meenakshi Krishnan, Vaughn

Osterman, Helio Perez-Stark, Vasanth Pidaparthy, Faranguisse Sadrieh, Stratos Tsoukanis, Trevor Turchan, Renata Valieva, Maeve Wildes, and Lixin Zheng.

I owe my deepest thanks and appreciation to all of my roommates over my graduate school years: Sam Bachhuber, Luke Bouck, Jesus Emilio Dominguez Russell, Nicholas Hiebert-White, Connor Lockhart, and Javier Reyes. You have kept me sane, provided me with copious amounts of entertainment and support, and I don't know what I would have done without any of you. You are all some of my dearest friends.

Nothing I have accomplished in my life would have been possible without my family. My mom and dad were my earliest supporters, and their unconditional love has helped me through every struggle. My aunt, Michelle, and uncle, Al, have acted as my spare parents (i.e., "sparents") during my years in Maryland, and they have kept me both entertained and well-fed. My grandparents, Katy, Paul, Diane, and Tom, have always been there for me with love and care, and they are among my favorite people in the world. I wrote much of this dissertation while sitting next to my Grandma Katy, and I wouldn't trade that time for anything. My "spare grandmother" Virginia bought me much of my University of Maryland merchandise and supported me during my time here; she will always be missed. My sister, Sarah, and my many aunts, uncles, and cousins have also supported me throughout my life, and I can't begin to express my gratitude for all of them. I love you all.

Finally, thank you to the fantasy author, Brandon Sanderson, who helped rekindle my love of reading.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
Arc.	Architecture
BS	Batch Size
CD	Cosine Distance
CNN	Convolutional Neural Network
CSTT	Channels Separated, Trained Together
CVPR	Conference on Computer Vision and Pattern Recognition
DFT	Discrete Fourier Transform
DS	Dilation-Separate
Ep.	Epochs
FFN	Feed Forward Network
FST	Fourier Scattering Transform
GAN	Generative Adversarial Network
GB	Giga-Byte
GELU	Gaussian Error Linear Unit
GFST	Generalized Fourier Scattering Transform
GPU	Graphics Processing Unit
HSI	Hyperspectral Imaging/ Image
ICASSP	International Conference on Acoustics, Speech, and Signal Processing
IDFT	Inverse Discrete Fourier Transform
Inv.	Inverse
LR	Learning Rate
LSTS	Layers Separated, Trained Separate
LSTT	Layers Separated, Trained Together
MAC	Multiply ACcumulate
MAE	Mean Absolute Error
MISR	Multi-Image SuperResolution
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MSI	MultiSpectral Image
MST++	Multi-stage Spectral-wise Transformer
Mtc.	Matching
NDT	No-Dilation-Together
NIR	Near-InfraRed
NTIRE	New Trends in Image Restoration and Enhancement
PCA	Principal Component Analysis
ReLU	RectiLinear Unit
RGB	Red Green Blue
SAM	Spectral Angle Mapper

SCP	Strong Complement Property
SSIM	Structural Similarity Index Measure
ST	Scattering Transform
SWI	Short-Wave Infrared
V	Validation
VIS	VISual range
VRAM	Video Random Access Memory
WD	Weight Decay
WST	Wavelet Scattering Transform

List of Notations

$\mathbb{1}_{\mathcal{X}}$	The indicator function of a set $\mathcal{X}$
$(a, b)_i$	A convolutional layer in layer i of a neural network with filter size a and dilation step b
A	Lower (semi-discrete) frame bound
$A_{\mathcal{S}}$	Lower (semi-discrete) frame bound for the filters indexed by $\mathcal{S}$
$A(\lambda)$	The set $\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})$ for $R > 0$, $N \in \mathbb{N}$, and $\xi_{\lambda,n} \in \mathbb{R}^d$ for all n
$\mathbf{A}$	An invertible linear transform from $\mathbb{R}^d$ to $\mathbb{R}^d$
$\mathcal{A}$	A countable subset of $\mathbb{R}^d$
α	Constant which controls the admissibility of a semi-discrete wavelet frame
B	Upper (semi-discrete) frame bound
$B_{k,\delta}$	The cube $\prod_{i=1}^{d} (\delta(k_i - 1), \delta k_i]$, for $\delta > 0$ and $k \in \mathbb{Z}^d$
$B_R(x)$	The ball centered at $x \in \mathbb{R}^d$ of radius R
$\mathcal{B}(\mathcal{X})$	The Borel sets of a topological space $\mathcal{X}$ (typically $\mathbb{R}^d$ or $\mathbb{C}$)
β_i	Hyperparameters for the Adam optimizer which control gradient computations, for $i \in \{1, 2\}$
$\widetilde{c}$	The counting measure
$\mathrm{cd}(h_{m,n}, \widetilde{h}_{m,n})$	The cosine distance of real spectra $h_{m,n}$ and predicted spectra $\widetilde{h}_{m,n}$
$\mathrm{cd}_\delta(h_{m,n}, \widetilde{h}_{m,n})$	A renormalized cd that avoids divide-by-0 errors
$\widehat{C}$	The number of channels of an image
C_K	A sequence that controls the rate of decay of the k-th layer energy of a GFST
$C^1(\mathbb{R}^d; \mathbb{R}^d)$	The space of continuously differentiable functions from $\mathbb{R}^d$ to $\mathbb{R}^d$
$C_{k_1,\ldots,k_n}(A_m)$	Cylindrical set for $\mathcal{X}_n$ with Borel set $A_m \in \mathcal{B}(\mathbb{C}^m)$
$\widetilde{C}_{t_1,\ldots,t_l}(A_l)$	Cylindrical set of functions from $\mathbb{R}^d$ to $\mathbb{C}$ with Borel set $A_l \in \mathcal{B}(\mathbb{C}^l)$
$\widetilde{C}$	The smallest σ-algebra generated by all $\widetilde{C}_{t_1,\ldots,t_l}(A_l)$
$\mathcal{C}_n$	The smallest σ-algebra generated by all $C_{k_1,\ldots,k_n}(A_m)$
$\mathrm{CD}(H, \widetilde{H})$	The average cosine distance $\mathrm{cd}(h_{m,n}, \widetilde{h}_{m,n})$ over all m, n
$\mathrm{CD}_\delta(H, \widetilde{H})$	The average cosine distance $\mathrm{cd}_\delta(h_{m,n}, \widetilde{h}_{m,n})$ over all m, n
D	The largest magnitude of an element of a set
$\widetilde{D}$	A constant that controls a decay rate for a generalized Fourier scattering transform
$\mathfrak{D}$	The discriminator network of a GAN
$\mathcal{D}_{n,\delta}$	The set of scaled dyadic integers of the form $\frac{\delta k}{2^n}$ for $k \in \mathbb{Z}^d$
$\mathcal{D}_\delta$	The set of dyadic integers scaled by $\delta > 0$
$\mathbb{E}[Y]$	Expectation of a random variable Y
$\mathbb{E}\left[\lvert S[\mathcal{P}]X(\cdot, t)\rvert^2\right]$	The $l^2L^2(\mathcal{P}; \Omega)$ norm of $S[\mathcal{P}]X(\cdot, t)$
f_M	A cutoff function on $\mathbb{C}$ of bound $M > 0$
f^*	The involution $f^*(x) \coloneqq \overline{f(-x)}$
$f * g(x)$	Convolution of functions f, g
$\widehat{f}(\xi)$	Fourier transform of a function f

$\check{f}(x)$	Inverse Fourier transform of a function f
F	An image
F_c	The c-th channel of an image F
$\widetilde{F}$	An image predicted by a machine learning model
$F \star g$	The periodic convolution of image F and filter g
$F \star_k w(l)$	The k-dilated convolution (actually cross-correlation) of image F with window w
$F_{Z_1,\ldots,Z_n}(a)$	Joint distribution function of random variables $Z_1, \ldots, Z_n : \Omega \to \mathbb{C}$
$\mathcal{F}$	Event space of a probability space
$\mathscr{F}$	A function from Λ^* to $\mathbb{C}$
G	A (finite) subgroup of $O(d)$
G^+	The quotient group $G/\{I, -I\}$ for G a finite subgroup of $O(d)$
$\mathfrak{G}$	The generator network of a GAN
$\mathcal{G}$	A semi-discrete Bessel sequence or semi-discrete frame
$\mathcal{G}_F$	A uniform covering frame
$\mathcal{G}_{F,\varepsilon}$	An ε-uniform covering frame
$\mathcal{G}_J$	A sequence of semi-discrete Bessel sequences
$\mathcal{G}_{F,J}$	A sequence of uniform covering frames
$\mathcal{G}_{W,J}$	A semi-discrete wavelet frame with indexing set $\Lambda_{W,J}$
γ	A lower bi-Lipschitz bound
$h_{m,n}$	A channel of a ground truth hyperspectral image
$\widetilde{h}_{m,n}$	A channel of a predicted hyperspectral image
H	A ground truth hyperspectral image
$\widetilde{H}$	A predicted hyperspectral image
$\mathcal{H}$	A sum of Gaussian translates
$\mathscr{H}$	A sum of Gaussian translates times indicator functions
I	Identity matrix on $\mathbb{R}^d$
J	Parameter controlling the number of dilations for a WST (continuous or discretized)
$k_\tau(\omega, t, s)$	A random kernel, depending on strictly stationary process τ
K_τ	A random integral operator with random kernel $k_\tau(\omega, t, s)$
$L^p(\mathbb{R}^d)$	Lebesgue space of functions on $\mathbb{R}^d$, for $1 \leq p \leq \infty$
$\\| \cdot \\|_{L^p}$	The L^p norm
$\\| \cdot \\|_{L^p/\sim}$	The quotient L^p norm determined by equivalence relation $\sim$
$l^2L^2(\mathcal{P}; \mathbb{R}^d)$	Space of sequences (with index $\mathcal{P}$) of functions on $\mathbb{R}^d$ with finite l^2L^2 norm
$\\| \cdot \\|_{l^2L^2}$	The l^2L^2 norm
$\\| \cdot \\|_{l^2L^2(\mathcal{S})}$	The l^2L^2 norm over the set $\mathcal{S} \subseteq \Lambda^*$
Λ	A countable indexing set
Λ_F	A countable indexing sets for an FST

$\Lambda_{F,\varepsilon}$	A countable indexing set for a GFST
Λ^*	The topological space $\Lambda^* \coloneqq (\Lambda \cup \{0\}) \times \mathbb{R}^d$, where $\Lambda \cup \{0\}$ has the discrete topology
Λ^k	The k-times Cartesian product of Λ
Λ_J	A sequence of indexing sets, indexed by J
$\Lambda_{F,J}$	A sequence of indexing sets for an FST
$\Lambda_{W,J}$	The indexing set $\{\lambda = (2^j, r) \in 2^{\mathbb{Z}} \times G : 2^j > 2^{-J}\}$ for WSTs
Λ_J^k	The k-time Cartesian product of Λ_J
m	Lebesgue measure on $\mathbb{R}^d$
M	The height of an image
M_h	The window height of the discretized FST filters
$M_\xi f$	The modulation of a function f by parameter $\xi \in \mathbb{R}^d$
$\mathcal{M}_h$	The window height of the discretized GFST filters
$\lvert \cdot \rvert$	Modulus or absolute value
μ	A bounded Borel measure on $\mathbb{R}^d$
μ_H	The mean of an image H
μ_p	The bounded Borel measure associated to $U[p]X$
$\mu_{F,p}$	The bounded Borel measure associated to $U_F[p]X$
μ_X	The bounded Borel measure associated to a strictly stationary process X
$\widetilde{\mu}$	The bounded Borel measure associated to $S[\mathcal{P}]X$
N	The width of an image
N_i	Hyperparameters for the MST++ network, where $i \in \{1, 2, 3, S\}$
N_w	The window width of the discretized FFST filters
$\mathcal{N}_w$	The window width of the discretized GFST filters
ν	A unimodular constant in $\mathbb{C}$
∇	Gradient with respect to Euclidean coordinates
$O(d)$	Orthogonal group of $\mathbb{R}^d$
Ω	The sample space of a probability space
$(\Omega, \mathcal{F}, P)$	A probability space
p	A path
P	A probability measure
P_k	Pooling operation for the k-th layer of a scattering transform
$\mathcal{P}$	The set of all paths contained in some Λ^k
$\mathcal{P}_F$	The set of all paths for a uniform covering frame
$\mathcal{P}_{F,\varepsilon}$	The set of all paths for an ε-uniform covering frame
$\mathcal{P}_J$	The set of all paths contained in some Λ_J^k
$\mathcal{P}_{F,J}$	The set of all paths contained in some $\Lambda_{F,J}^k$
$\mathcal{P}_{W,J}$	The set of all paths contained in some $\Lambda_{W,J}^k$
ϕ	A father wavelet
ϕ_J	A dilated father wavelet
Φ	The distribution function of a standard Gaussian

φ	Auxiliary functions used in proofs
ψ	A mother wavelet
$\psi_{j,r}$	A dilated and rotated mother wavelet
Ψ	Function used in the admissibility condition for a semi-discrete wavelet frame
$\widetilde{\Psi}$	Convolution operator from $\mathcal{X}_n$ to $\{h : \mathbb{R}^d \to \mathbb{C}\}$
$\widetilde{\Psi}_t$	Coordinate of the operator $\widetilde{\Psi}$ for $t \in R^d$
Q	Parameter controlling the number of rotations for a discretized WST
$Q_R(x)$	A cube centered at $x \in \mathbb{R}^d$ of side length $2R$
r_{θ_q}	Rotation in the plane by angle θ_q
$R_X(s)$	Autocorrelation function of a strictly stationary process X
$R_p(s)$	The autocorrelation function of X_p
$R_{S[\mathcal{P}]X}(s)$	The autocorrelation function of $S[\mathcal{P}]X$
$\text{ReLU}(x)$	The function $\max\{0, x\}$ for $x \in \mathbb{R}$
ρ	Function used in the admissibility condition for a semi-discrete wavelet frame
$\text{sam}(h_{m,n}, \widetilde{h}_{m,n})$	The SAM metric of real spectra $h_{m,n}$ and predicted spectra $\widetilde{h}_{m,n}$
$\text{sam}_\delta(h_{m,n}, \widetilde{h}_{m,n})$	A renormalized sam that avoids divide-by-0 errors
S	The number of scattering channels of a discretized scattering transform
$S[J, Q]F$	The discretized (2-layer) WST of an image F
$S_3[J, Q]F$	The discretized 3-layer WST of an image F
$S[M_h, N_w]F$	The discretized FST of an image F
$S[\mathcal{M}_h, \mathcal{N}_w]F$	The discretized GFST of an image F
$S[\mathcal{P}]f(x)$	The scattering transform of a function f
$S[\mathcal{P}_F]f(x)$	The Fourier scattering transform of a function f
$S[\mathcal{P}_{F,\varepsilon}]f$	The generalized Fourier scattering transform of a function f
$S[\mathcal{P}_{W,J}]f(x)$	The wavelet scattering transform of a function f
$S[\mathcal{P}]X(\omega, t)$	The stationary scattering transform of a strictly stationary process X
$S[\mathcal{P}_F]X(\omega, t)$	The stationary Fourier scattering transform of a strictly stationary process X
$\mathcal{S}$	A measurable subset of $\Lambda \times \mathbb{R}^d$
$\mathcal{S}(\mathbb{R}^d)$	The space of Schwartz functions on $\mathbb{R}^d$
$\text{SAM}(H, \widetilde{H})$	The average SAM $\text{sam}(h_{m,n}, \widetilde{h}_{m,n})$ over all m, n
$\text{SAM}_\delta(H, \widetilde{H})$	The average SAM $\text{sam}_\delta(h_{m,n}, \widetilde{h}_{m,n})$ over all m, n
$\text{ssim}(H_c, \widetilde{H}_c)$	The ssim between actual channel H_c and predicted channel $\widetilde{H}_c$
$\text{SSIM}(H, \widetilde{H})$	The average $\text{ssim}(H_c, \widetilde{H}_c)$ over all channels c
$\text{supp}(f)$	The support of a function f
$\sigma_{\mathcal{G}}$	The largest $\sigma \geq 0$ such that a set $\mathcal{G}$ satisfies the σ-strong complement property
σ_H	The standard deviation of an image H
$\sigma_{H\widetilde{H}}$	The covariance of images H and $\widetilde{H}$
σ_k	Pointwise activation function for the k-th layer of a scattering transform

$T_c f$	Translation of function f by parameter $c \in \mathbb{R}^d$
$T_\tau f$	Translation of function f by deformation $\tau \in C^1(\mathbb{R}^d; \mathbb{R}^d)$
$\mathcal{T}$	The set $(\Lambda \times \mathbb{R}^d) \setminus \mathcal{S}$, for a measurable subset $\mathcal{S} \subseteq \Lambda \times \mathbb{R}^d$
$\tau(x)$	An element of $C^1(\mathbb{R}^d; \mathbb{R}^d)$
$\tau(\omega, t)$	A strictly stationary process from $\Omega \times \mathbb{R}^d$ to $\mathbb{R}^d$
θ_q	The angle $\frac{\pi q}{Q}$
$U[p]f(x)$	The scattering propagator of a function f for the path $p \in \mathcal{P}$
$U_F[p]f(x)$	The Fourier scattering propagator of a function f for the path $p \in \mathcal{P}_F$
$U_{F,\varepsilon}[p]f(x)$	The generalized Fourier scattering propagator of a function f for the path $p \in \mathcal{P}_{F,\varepsilon}$
$U_{W,J}[p]f$	The wavelet scattering propagator of a function f for the path $p \in \mathcal{P}_{W,J}$
$U_J[p]f(x)$	The scattering propagator of a function f for the path $p \in \mathcal{P}_J$
$\{U[p]f : p \in \Lambda^k\}$	k-th layer of a scattering propagator
$\{U[p]f * g_0 : p \in \Lambda^k\}$	k-th layer of a scattering transform
$\sum_{p \in \Lambda^k} \|U[p]f\|_{L^2}^2$	Energy of k-th layer of a scattering propagator
$U[p]X(\omega, t)$	The stationary scattering propagator of a strictly stationary process X for the path $p \in \mathcal{P}$
$U_F[p]X(\omega, t)$	The Fourier stationary scattering propagator of a strictly stationary process X for the path $p \in \mathcal{P}_F$
$\{U[p]X : p \in \Lambda^k\}$	k-th layer of a stationary scattering propagator
$\{U[p]X * g_0 : p \in \Lambda^k\}$	k-th layer of a stationary scattering transform
$\sum_{p \in \Lambda^k} \mathbb{E}\left[\|U[p]X(\cdot, t)\|^2\right]$	Energy of k-th layer of a stationary scattering propagator
$X(\omega, t)$	A strictly stationary random process on $\Omega \times \mathbb{R}^d$
$X_p(\omega, t)$	The strictly stationary process $U[p]X(\omega, t)$
$X_{\{\delta\}}(\omega, t)$	A simple strictly stationary process on $\Omega \times \mathbb{R}^d$ with parameter $\delta > 0$
$X_\tau(\omega, t)$	The random process equal to $X(\omega, t - \tau(\omega, t))$ for strictly stationary processes τ, X
$X * g(\omega, t)$	Convolution of a strictly stationary random process X and a function g
$\mathcal{X}_n$	The set of M-bounded functions on $\mathbb{R}^d$ that are constant on each $B_{k,\delta}$

Chapter 1: Introduction

Modern artificial neural network techniques have seen immense success over the last two decades in a wide variety of applications. In particular, convolutional neural networks (CNNs) have repeatedly been shown to learn complicated features of the input data for tasks like classification and data generation. However, the success of these networks is dependent on being able to effectively train them to minimize some loss function, which often requires access to extensive computational resources and large amounts of training data. At the same time, the fact that the convolutional filters in CNNs are not known a priori can make it difficult to prove interesting and insightful results about these networks.

In part to overcome these difficulties, much attention has been devoted to the study of scattering transforms. These are mathematical operators from harmonic analysis which are similar in structure to CNNs but with predefined filters with known mathematical structure. This structure has allowed for the proof of several different properties of scattering transforms, including that they are bounded and Lipschitz continuous nonlinear operators which are stable with respect to small deformations, and that, in some cases, they exhibit fast energy decay. Much of the study of these objects has focused on the case when the filters are constructed from directional wavelets, though time-frequency based filters and more general families have also been considered.

Beyond mathematical understanding, scattering transforms, and their pre-defined structures, also

have important uses in applications. Rather than needing to spend computational resources and time to train CNNs to extract features from the data, one can instead use different choices of scattering transforms with the knowledge that particular features will be identified and extracted. These features can be combined with classical machine learning techniques or other neural networks which are then more easily able solve the task at hand by making use of these features (as opposed to only the raw data). Scattering transforms have seen impressive performance for a variety of image and audio classification and data generation tasks.

Despite their proven track record, however, there are still many aspects of scattering transforms that are not well-understood. While non-wavelet based scattering transforms have been constructed, many results and extensions of wavelet scattering transforms have not been studied for these more general classes of filters. Additionally, other potential properties of these transforms, such as invertibility, have received little attention, despite their potential implications for applications where one might want to reconstruct the data from the extracted features.

This dissertation is organized as follows. In Chapter 2, we present preliminary and background information on several topics that will be discussed throughout this dissertation. In particular, we define scattering transforms, state several of their properties, and discuss two particular examples. We also define and state several properties of strictly stationary random processes, and we discuss an approximate numerical inverse of wavelet scattering transforms. We end with a discussion of hyperspectral images and the hyperspectral reconstruction problem.

In Chapter 3, we extend the result on the limiting translation invariance behavior of wavelet scattering transforms to more general sequences of scattering transforms.

In Chapter 4, we extend a previous construction of wavelet scattering transforms for strictly stationary random process to use general families of convolutional filters. We prove the corresponding

properties for these transforms in full generality, as well as in the particular case of stationary Fourier scattering transforms, which use time-frequency based filters.

In Chapter 5, we discuss applications of wavelet scattering transforms to the hyperspectral reconstruction problem, in which hyperspectral images are generated from smaller and easier-to-obtain images. We propose a novel pipeline called Hyperscat, which uses a neural network to match the scattering coefficients of the input images to the coefficients of the corresponding hyperspectral images, followed by another neural network which inverts the scattering embedding. We discuss our many results from experiments designed to optimize our reconstructions, and we show that our pipeline achieves comparable results to a much more complicated neural network, but with improved computational efficiency, on both the Hyper-Skin and ARAD datasets.

Finally, in Chapter 6, we study methods for inverting scattering transforms which are similar in structure to Fourier scattering transforms. First, we construct generalizations of Fourier scattering transforms which have the same properties but are invertible in some cases. Next, we study global stability properties of inverting scattering transforms in the continuous setting. Finally, we study numerical inversions of both standard and generalized Fourier scattering transforms, and we compare the inversion results to the wavelet case.

Chapter 2: Preliminaries

2.1 Notation

In this section, we briefly introduce some notation that we will use throughout this dissertation.

For $x, y \in \mathbb{R}^d$, let $x \cdot y$ denote the Euclidean inner product and $|x|$ the Euclidean norm. For $z \in \mathbb{C}$, we only slightly abuse this notation when we use $|z|$ for the modulus, and we write $\mathrm{Re}(z)$ to be the real part of z and $\mathrm{Im}(z)$ to be the imaginary part. For $R > 0$, $x \in \mathbb{R}^d$, we define the ball $B_R(x) \coloneqq \left\{y \in \mathbb{R}^d : |y - x| < R\right\}$ and the cube $Q_R(x) \coloneqq \{y \in \mathbb{R}^d : |y_i - x_i| < R, i = 1, \ldots, n\}$.

Let Ω be a set and $\mathcal{F}$ a σ-algebra of subsets of Ω (see [20]). We say a σ-additive set function $P : \mathcal{F} \to [0, \infty]$ is a measure and we say $(\Omega, \mathcal{F}, P)$ is a measure space. If $P(\Omega) = 1$, we say P is a probability measure and $(\Omega, \mathcal{F}, P)$ is a probability space. For $A_\Omega \in \mathcal{F}$, we say that A_Ω occurs P-almost surely, or P-a.s., if $P(\Omega \setminus A_\Omega) = 0$. If $Z : \Omega \to \mathbb{C}$ is a random variable, we denote its expectation, the integral with respect to the measure P, by $\mathbb{E}[Z]$.

If $\mathcal{X}$ is a topological space, we let $\mathcal{B}(\mathcal{X})$ denote the Borel σ-algebra generated by the open subsets of $\mathcal{X}$. We call a measure $\mu : \mathcal{B}(\mathcal{X}) \to [0, \infty]$ a Borel measure, and $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$ is a Borel measure space. We say a measure μ is bounded if $\mu(\mathcal{X}) < \infty$. In the special case that $\mathcal{X} = \mathbb{R}^d$ or $\mathbb{C}$, we denote the measure as m, the Lebesgue measure. For $A_\mathcal{X} \in \mathcal{B}(\mathcal{X})$, we say that $A_\mathcal{X}$ occurs μ-almost everywhere, or μ-a.e., if $\mu(\mathcal{X} \setminus A_\mathcal{X}) = 0$.

For a function $f : \mathbb{R}^d \to \mathbb{C}$ (or $f : \mathbb{R}^d \to \mathbb{R}^d$), we denote $\|f\|_{L^p}^p \coloneqq \int\limits_{\mathbb{R}^d} |f|^p \mathrm{d}x$ for $1 \leq p < \infty$ (where we use $\mathrm{d}x$ in place of the Lebesgue measure $\mathrm{d}m$) and $\|f\|_{L^\infty} \coloneqq \inf\{T > 0 : m(|f| \geq T) = 0\}$. For $1 \leq p \leq \infty$, we say $f \in L^p(\mathbb{R}^d)$ if $\|f\|_{L^p} < \infty$. Given a countable indexing set $\mathcal{P}$, we say the sequence $\{f_p\}_{p\in\mathcal{P}} \in l^2L^2(\mathcal{P};\mathbb{R}^d)$ if $\|\{f_p\}_p\|_{l^2L^2}^2 \coloneqq \sum\limits_{p\in\mathcal{P}} \|f_p\|_2^2 < \infty$. We say a function $\tau : \mathbb{R}^d \to \mathbb{R}^d$ is $C^1(\mathbb{R}^d;\mathbb{R}^d)$, i.e., $\tau \in C^1(\mathbb{R}^d;\mathbb{R}^d)$, if τ is continuously differentiable on $\mathbb{R}^d$.

For a function $f \in L^1\left(\mathbb{R}^d\right) \cap L^2\left(\mathbb{R}^d\right)$, we define its Fourier transform

$$\widehat{f}(\xi) \coloneqq \int_{\mathbb{R}^d} f(x) e^{-2\pi i x\cdot\xi} \mathrm{d}x. \tag{2.1}$$

We can extend this definition in the usual way as an isometry on $L^2\left(\mathbb{R}^d\right)$ [19]. In that case, the inverse Fourier transform is the isometric extension of the operator

$$\check{f}(x) \coloneqq \int_{\mathbb{R}^d} f(\xi) e^{2\pi i \xi\cdot x} \mathrm{d}\xi. \tag{2.2}$$

Next, for $f, g : \mathbb{R}^d \to \mathbb{C}$ we define their convolution $f * g : \mathbb{R}^d \to \mathbb{C}$ by

$$f * g(x) = \int_{\mathbb{R}^d} f(y) g(x-y) \mathrm{d}y. \tag{2.3}$$

Recall by Young's Convolutional Inequality that if $f \in L^p(\mathbb{R}^d)$ for some $1 \leq p \leq \infty$ and $g \in L^1(\mathbb{R}^d)$, then $f * g \in L^p(\mathbb{R}^d)$, with $\|f * g\|_{L^p} \leq \|f\|_{L^p}\|g\|_{L^1}$. If $c \in \mathbb{R}^d$ or $\xi' \in \mathbb{R}^d$, we define the translation operator T_c and modulation operator $M_{\xi'}$ by $T_c f(x) = f(x-c)$ and $M_\xi f(x) = e^{2\pi i \xi'\cdot x} f(x)$.

Let $\mathcal{S}(\mathbb{R}^d)$ be the set of Schwartz functions on $\mathbb{R}^d$, i.e., functions $f : \mathbb{R}^d \to \mathbb{C}$ such that, for all

$\alpha, \beta \in \mathbb{Z}^d$ with $\alpha_i, \beta_i \geq 0$ for all $i = 1, \ldots, d$,

$$\sup_{x \in \mathbb{R}^d} \left| x_1^{\alpha_1} \cdots x_d^{\alpha_d} (\partial_1^{\beta_1} \cdots \partial_d^{\beta_d} f)(x) \right| < \infty.$$

For a function $f : \mathbb{R}^d \to \mathbb{C}$, we define its support, $\text{supp}(f) := \overline{\{x \in \mathbb{R}^d : f(x) \neq 0\}}$, where $\overline{A}$ denotes the topological closure of a set A. We say $f \in L^2(\mathbb{R}^d)$ is R-bandlimited for some $R > 0$ if $\text{supp}(\widehat{f}\,) \subseteq Q_R(0)$, and we say f is bandlimited if $\widehat{f}$ has compact support. We say that f is (ε, R)-bandlimited for $R > 0$ and $\epsilon \in [0, 1)$ if $(1 - \varepsilon)\|f\|_{L^2} \leq \left\|\widehat{f}\right\|_{L^2(Q_0(R))}$. Note that, by the Plancherel Theorem, any function $f \in L^2$ is (ε, R)-bandlimited for some pair R, ε.

2.2 Scattering Transforms

Since they were first constructed by Stéphane Mallat in 2012 [88], scattering transforms have proven to be extremely valuable tools for data analysis. Scattering transforms act as "feature extractor", which are tools for finding and organizing the most fundamental descriptors (or features) of the input data for the task at hand, such as classification. These transforms consist of repeated applications, or *layers*, of convolutional filters and nonlinear, pointwise operators. They are thus similar in structure to convolutional neural networks (CNNs), one of the main tools in modern machine learning [76], and it has been argued that scattering transforms can be viewed as a mathematical model for general CNNs [89].

However, scattering transforms differ from CNNs in a few key respects. First, the filters in CNNs are "trained", i.e., iteratively changed, so that the whole model minimizes some loss function. While this allows CNNs to be optimized for a desired problem, this comes with the tradeoffs of needing significant time and computational resources for accurate training [131], as well as large amounts

of training data [8]. Scattering transforms, on the other hand, use a priori selected filters with known mathematical structure, thus eliminating the need for training and also making it easier to find provable guarantees. Additionally, while CNNs are designed to work with discrete data, scattering transforms can act in both discrete and continuous settings. Finally, while only the outputs of the final layer of CNNs are used, the scattering transform provides the outputs of every layer.

To deal with different types of data, numerous kinds of scattering transforms have been created. Mallat's original scattering transform, constructed with the help of Joan Bruna, used directional wavelet-based convolutional filters to capture scale and direction information about the data, as well as complex modulus as the nonlinearity [25, 26, 88]. It has since been generalized by Wojciech Czaja and Weilin Li to use time-frequency-based filters (for capturing location and frequency information) [44, 45, 79], while Helmut Bölcskei and Thomas Wiatowski constructed versions with more general, frame-based filters as well as general nonlinear filters [121, 122]. While these are all built to act on functions on Euclidean space, various researchers have also constructed scattering transform to act on data from stochastic processes [24,26,88], graphs [31,54,104,105,137], and manifolds [53,104]. As feature extractors, scattering transforms have seen use in a wide array of classification tasks for audio [9–12, 86, 108], images [17, 24, 26, 71, 96–98, 102, 128], and 3-dimensional data [50, 63, 70, 71, 80].

In this section, we will first define semi-discrete Bessel sequences and semi-discrete frames, which will be the collections of convolutional filters we consider in this work. Then, we will define scattering transforms and consider two prominent examples: the wavelet scattering transform and the Fourier scattering transform. Finally, we will state several properties of these transforms, both in the general case and for these two noted examples.

2.2.1 Semi-Discrete Bessel Sequences and Frames

First, we consider the convolution filters that will be used in defining the scattering transform (cf. Definition 6 in [121]).

Definition 2.2.1. *Let Λ be a countable indexing set such that $0 \notin \Lambda$. Consider a sequence $\mathcal{G} := \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$.*

1. *We say $\mathcal{G}$ is a semi-discrete Bessel sequence if there exists a constant $0 < B < \infty$ such that*

$$\|f * g_0\|_{L^2}^2 + \sum_{\lambda \in \Lambda} \|f * g_\lambda\|_{L^2}^2 \leq B\|f\|_{L^2}^2 \text{ for all } f \in L^2(\mathbb{R}^d). \tag{2.4}$$

2. *We say $\mathcal{G}$ is a semi-discrete frame if there exists constants $0 < A \leq B < \infty$ such that*

$$A\|f\|_{L^2}^2 \leq \|f * g_0\|_{L^2}^2 + \sum_{\lambda \in \Lambda} \|f * g_\lambda\|_{L^2}^2 \leq B\|f\|_{L^2}^2 \text{ for all } f \in L^2(\mathbb{R}^d). \tag{2.5}$$

If $A = B = 1$, we say $\mathcal{G}$ is a semi-discrete Parseval frame.

Note that, by the Plancherel Theorem, $\mathcal{G}$ is a semi-discrete Bessel sequence if and only if

$$|\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda} |\widehat{g_\lambda}(\xi)|^2 \leq B \text{ for all } \xi \in \mathbb{R}^d \tag{2.6}$$

and similarly $\mathcal{G}$ is a semi-discrete frame if and only if

$$A \leq |\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda} |\widehat{g_\lambda}(\xi)|^2 \leq B \text{ for all } \xi \in \mathbb{R}^d. \tag{2.7}$$

Note that these equations hold for all $\xi \in \mathbb{R}^d$ (rather than for m-a.e. $\xi \in \mathbb{R}^d$) because $\widehat{g_0}, \widehat{g_\lambda}$ are

continuous since $g_0, g_\lambda \in L^1(\mathbb{R}^d)$, for all $\lambda \in \Lambda$. Thus we may equivalently define semi-discrete Bessel sequences (respectively semi-discrete frames) using equation 2.6 (respectively equation 2.7).

To understand why the sets $\mathcal{G}$ are referred to as *semi-discrete*, first denote $f^*(x) = \overline{f(-x)}$ for any $f \in L^2(\mathbb{R}^d)$. Then, note that we can write $f * g_\lambda$ as

$$f * g_\lambda(x) = \int_{\mathbb{R}^d} f(y) g_\lambda(x-y) \mathrm{d}y = \int_{\mathbb{R}^d} f(y) \overline{g_\lambda^*(y-x)} \mathrm{d}y = \int_{\mathbb{R}^d} f(y) \overline{T_x g_\lambda^*(y)} \mathrm{d}y = \langle f, T_x g_\lambda^* \rangle_{L^2}. \quad (2.8)$$

Thus, the values $f * g_\lambda(x)$ are the (standard) frame coefficients of f for the element $T_x g_\lambda^*$, where $x \in \mathbb{R}^d$ and $\lambda \in \Lambda \cup \{0\}$ are arbitrary. We can also rewrite Equation 2.4 as

$$\int_{\mathbb{R}^d} \sum_{\lambda \in \Lambda} |\langle f, T_x g_\lambda^* \rangle|^2 \mathrm{d}x \leq B \|f\|_{L^2}^2 \quad (2.9)$$

and similarly for Equation 2.5. Hence, similar to the case of standard discrete Bessel sequences (or frames), we say that $\{T_x g_\lambda^* : x \in \mathbb{R}^d, \lambda \in \Lambda \cup \{0\}\}$ is a semi-discrete Bessel sequence (or frame), with discrete parameter λ and continuous parameter x.

2.2.2 Scattering Transforms Definitions

Next, we will define scattering transforms, whose filters are semi-discrete Bessel sequences. For the rest of this dissertation, we shall always assume that the upper frame bounds of our sequences of filters satisfy $B \leq 1$.

For $k \in \mathbb{N}$, define $\Lambda^k = \overbrace{\Lambda \times \cdots \times \Lambda}^{k \text{ times}}$; also, define $\Lambda^0 = \{\varnothing\}$. We refer to $p = (\lambda_1, \ldots, \lambda_k) \in \Lambda^k$ as a *path* and we let $\mathcal{P} := \bigcup_{k=0}^{\infty} \Lambda^k$ be the set of all paths. For two paths $p, p' \in \mathcal{P}$, we define their concatenation by $(p, p') = (\lambda_1, \ldots, \lambda_k, \lambda'_1, \ldots, \lambda'_{k'})$.

To define the scattering transform, we first define the scattering propagator (cf. Definition 2 in [121]).

Definition 2.2.2. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda}$ be a semi-discrete Bessel sequence. Given a path $p \in \Lambda^k$, the scattering propagator $U[p] : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ for $f \in L^2(\mathbb{R}^d)$ is inductively defined by*

$$U[p]f(x) = \begin{cases} f(x), & k = 0, \\ |U[(\lambda_1, \ldots, \lambda_{k-1})]f * g_{\lambda_k}(x)|, & k \geq 1. \end{cases} \tag{2.10}$$

Since each $g_\lambda \in L^1(\mathbb{R}^d)$, it is clear that each $U[p]$ is a well-defined operator. Now, we can define the scattering transform (cf. Definition 3 in [121]).

Definition 2.2.3. *For $f \in L^2(\mathbb{R}^d)$, the scattering transform $S[\mathcal{P}] : L^2(\mathbb{R}^d) \to l^2 L^2(\mathcal{P}; \mathbb{R}^d)$ is defined by*

$$S[\mathcal{P}]f(x) = \{U[p]f * g_0(x) : p \in \mathcal{P}\}. \tag{2.11}$$

As a consequence of Corollary 2.2.14, we will have that $\|S[\mathcal{P}]f\|_{l^2 L^2} < \infty$ if $f \in L^2(\mathbb{R}^d)$, and so $S[\mathcal{P}]$ is well defined.

Remark 2.2.4. *The scattering transforms defined here (Definitions 2.2.2 and 2.2.3) can be considered a somewhat less general version of the scattering transforms constructed by Wiatowski and Bölcskei in [121]. Indeed, the latter allows for the application of different sets of filter $\mathcal{G}_k$ for each $k \in \mathbb{N}$. Additionally, their scattering propagators also allowed for the application of different Lipschitz-continuous, nonlinear operators $M_k : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ instead of just modulus, as well as applications of additional Lipschitz operators $P_k : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ which modeled pooling operations found in neural networks. However, many of the results proven in this dissertation are easily extendable to this gener-*

ality (with some exceptions that will be explicitly noted); we leave the proofs to the interested reader.

Also, in [121], every set of filters $\mathcal{G}_k$ considered is a semi-discrete frame, rather than a semi-discrete Bessel sequence as in our definitions. It should be noted, though, that no result in [121] requires the set $\mathcal{G}_k$ to have a lower frame bound A_k, and so we have chosen to drop it from the general definition. This will not preclude us from considering $\mathcal{G}$ which are semi-discrete frames, as shall be seen in Section 2.2.3.

Before continuing, we shall give two small definitions that will be useful throughout this work. We shall refer to the sets $\{U[p] : p \in \Lambda^k\}$ and $\{U[p] * g_0 : p \in \Lambda^k\}$ for fixed k as *layers* of the scattering propagator and scattering transform, respectively. These are analogous to the layers of CNNs. Additionally, we shall refer to the function g_0 as the *output-generating function*.

2.2.3 Examples of Scattering Transforms

Next, we shall consider two particular examples of scattering transforms that are particularly notable in the literature: the wavelet scattering transform (WST), which was the original scattering transform constructed by Mallat in [88], and the Fourier scattering transform (FST), constructed by Czaja and Li [44].

2.2.3.1 Wavelet Scattering Transforms

Let $O(d)$ be the orthogonal group for $\mathbb{R}^d$, and let $G < O(d)$ be a finite subgroup such that $-I \in G$, where I is the identity matrix. Define $\Lambda_W \coloneqq 2^{\mathbb{Z}} \times G$. Fix $\phi, \psi \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$, and for

$\lambda = (2^j, r) \in \Lambda_W$, define $\psi_\lambda(x) \coloneqq 2^{dj}\psi(2^j r^{-1}x)$. Assume that, for all $\xi \in \mathbb{R}^d$, we have

$$\sum_{\lambda\in\Lambda_W} \left|\widehat{\psi_\lambda}(\xi)\right|^2 = \sum_{j=-\infty}^{\infty}\sum_{r\in G}\left|\widehat{\psi}(2^{-j}r^{-1}\xi)\right|^2 = 1 \tag{2.12}$$

and

$$|\widehat{\phi}(\xi)|^2 = \sum_{j=-\infty}^{0}\sum_{r\in G}\left|\widehat{\psi}(2^{-j}r^{-1}\xi)\right|^2. \tag{2.13}$$

We call ψ the mother wavelet, and ϕ the father wavelet.

For $J \in \mathbb{Z}$, define $\Lambda_{W,J} \coloneqq \left\{\lambda = (2^j, r) \in \Lambda_W : 2^j > 2^{-J}\right\}$ and $\phi_J(x) \coloneqq 2^{-dJ}\phi(2^{-J}x)$. Combining equations 2.12 and 2.13, we have for all $\xi \in \mathbb{R}^d$ that

$$\left|\widehat{\phi_J}(\xi)\right|^2 + \sum_{\lambda\in\Lambda_{W,J}}\left|\widehat{\psi_\lambda}(\xi)\right|^2 = 1. \tag{2.14}$$

Thus, for each $J \in \mathbb{Z}$, the set $\mathcal{G}_{W,J} = \{\phi_J\} \cup \{\psi_\lambda\}_{\lambda\in\Lambda_{W,J}}$ is a semi-discrete frame with frame bounds $A = B = 1$. We call $\mathcal{G}_{W,J}$ a semi-discrete wavelet frame.

With this, we can define the wavelet scattering transform.

Definition 2.2.5. (Definition 2.4 in [88]) *Let $\mathcal{G}_{W,J}$ be a semi-discrete wavelet frame, as defined above. Let $\mathcal{P}_{W,J} = \bigcup_{k=0}^{\infty}\Lambda_{W,J}^k$ be the set of paths. The wavelet scattering transform (WST) $S[\mathcal{P}_{W,J}] : L^2(\mathbb{R}^d) \to l^2L^2(\mathcal{P}_{W,J};\mathbb{R}^d)$ is a scattering transform with semi-discrete Bessel sequence $\mathcal{G}_{W,J}$ and wavelet scattering propagators $U_{W,J}[p]$ for $p \in \mathcal{P}_{W,J}$.*

Remark 2.2.6. *As explained in [88], if f and $\widehat{\psi}$ are real-valued then $f * \phi_{(2^j,-r)}(x) = \overline{f * \phi_{(2^j,r)}(x)}$. Hence, $U_{W,J}[(2^j,-r)]f = U_{W,J}[(2^j,r)]f$ for all $r \in G$, meaning that $U_{W,J}[\lambda]f$ only has to be calculated for $\lambda = (2^j, r)$ with $r \in G^+ \coloneqq G/\{I, -I\}$.*

Mallat also introduced an additional admissibility condition for ψ.

Definition 2.2.7. (See Theorem 2.6 of [88]) *A mother wavelet ψ is admissible if there exists $\eta \in \mathbb{R}^d$ and a function $\rho : \mathbb{R}^d \to \mathbb{C}$ satisfying $\rho(x) \geq 0$ for all $x \in \mathbb{R}^d$, $|\widehat{\rho}(\xi)| \leq |\widehat{\phi}(2\xi)|$ for all $\xi \in \mathbb{R}^d$, and $\widehat{\rho}(0) = 1$, such that the function*

$$\widehat{\Psi}(\xi) \coloneqq |\widehat{\rho}(\xi - \eta)|^2 - \sum_{k=1}^{\infty} k(1 - \left|\widehat{\rho}(2^{-k}(\xi - \eta))\right|^2)$$

satisfies

$$\alpha \coloneqq \inf_{1 \leq |\xi| \leq 2} \sum_{j=-\infty}^{\infty} \sum_{r \in G} \widehat{\Psi}(2^{-j} r^{-1} \xi) |\widehat{\psi}(2^{-j} r^{-1} \xi)|^2 > 0 \tag{2.15}$$

Mallat use this admissibility condition to prove several additional results about the WST. However, while Mallat provided an example of an admissible Battle-Lemarié wavelet in $d = 1$, no admissible wavelets are known in dimensions $d > 1$.

2.2.3.2 Fourier Scattering Transforms

In order to construct the FST, we first need to define uniform covering frames as in [44].

Definition 2.2.8. (Definition 2.1 in [44]) *Let Λ_F be a countable indexing set with $0 \notin \Lambda_F$, and let $\mathcal{G}_F = \{g_0\} \cup \{g_\lambda : \lambda \in \Lambda_F\}$.We say $\mathcal{G}_F$ is a uniform covering frame if it satisfies the following assumptions.*

1. *Assumptions on g_0 and g_λ. Let $g_0 \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be such that $\widehat{g_0}$ is supported in a neighborhood of the origin and $|\widehat{g_0}(0)| = 1$. For each $\lambda \in \Lambda_F$, let $g_\lambda \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be such that* $\text{supp}(\widehat{g_\lambda})$ *is compact and connected.*

2. *Uniform covering property. For any $R > 0$, there exists an integer $N > 0$ such that for each $\lambda \in \Lambda_F$, the set* $\text{supp}(\widehat{g_\lambda})$ *can be covered by N cubes of side length $2R$.*

3. *Frame condition. Assume that, for all $\xi \in \mathbb{R}^d$,*

$$|\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda_F} |\widehat{g_\lambda}(\xi)|^2 = 1. \tag{2.16}$$

Remark 2.2.9. *The uniform covering property is equivalent to saying that the supports of $\widehat{g_\lambda}$ are each contained in some cube of fixed side length $2R > 0$, where R is independent of λ.*

With this, we can define the Fourier scattering transform.

Definition 2.2.10. (Definition 2.4 in [44]) *Let $\mathcal{G}_F$ be a uniform covering frame with indexing set Λ_F. Let $\mathcal{P}_F = \bigcup_{k=0}^{\infty} \Lambda_F^k$ be the set of all paths. The Fourier scattering transform (FST) $S[\mathcal{P}_F] : L^2(\mathbb{R}^d) \to l^2 L^2(\mathcal{P}_F; \mathbb{R}^d)$ is a scattering transform with semi-discrete Bessel sequence $\mathcal{G}_F$ and scattering propagators $U_F[p]$ for $p \in \mathcal{P}_F$*

Czaja and Li provide a particular class of examples of uniform covering frames known as semi-discrete Gabor frames.

Proposition 2.2.11. (Proposition 2.3 from [44]) *Let $g \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a function such that* $\text{supp}(\widehat{g})$ *is compact and connected, $|\widehat{g}(0)| = 1$, and $\sum_{m \in \mathbb{Z}^d} |\widehat{g}(\xi - m)|^2 = 1$ for all $\xi \in \mathbb{R}^d$. Let $\mathbf{A} : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible, linear transformation. We let $\Lambda_F = \mathbf{A}^T \mathbb{Z}^d \setminus \{0\}$, $g_0(x) = |\det \mathbf{A}| g(\mathbf{A}x)$, and $g_\lambda(x) = e^{2\pi i \lambda \cdot x}$ for $\lambda \in \Lambda_F$. Then, the set $\mathcal{G}_F = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_F}$ is a semi-discrete Gabor frame and a uniform covering frame.*

2.2.4 Properties of Scattering Transforms

Next, we will note several important properties of scattering transforms. Recall that we have assumed that all semi-discrete Bessel sequences $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda}$ will have an upper frame bound satisfying $B \leq 1$.

First, we have that scattering transforms are Lipschitz-continuous, i.e., non-expansive.

Theorem 2.2.12. (Proposition 4 in [121]) *Let $S[\mathcal{P}]$ be a scattering transform. For all $f, g \in L^2(\mathbb{R}^d)$, we have*

$$\|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2 L^2} \leq \|f - g\|_{L^2}. \tag{2.17}$$

Remark 2.2.13. *More general Lipschitz results can be found in [18, 136], which considers both different B_k for each layer as well as $B_k > 1$. However, Lipschitz results only hold in that case under the assumption that either the scattering network has finite layers, i.e. $\mathcal{P} = \bigcup_{k=0}^{K} \lambda^k$ for some $K \in \mathbb{N}$, or that $B_1 \prod_{k=2}^{\infty} \max\{1, B_k\} < \infty$. For ease of presentation, we restrict ourselves to $B_k = B$ for all k and $B \leq 1$; we leave the extension of our results to the interested reader.*

As a corollary of Theorem 2.2.12 (i.e., taking $g \equiv 0$), we have that $S[\mathcal{P}]$ is a bounded operator.

Corollary 2.2.14. (Proposition 1 in [121]) *Let $S[\mathcal{P}]$ be a scattering transform. For all $f \in L^2(\mathbb{R}^d)$, we have*

$$\|S[\mathcal{P}]f\|_{l^2 L^2} \leq \|f\|_{L^2}. \tag{2.18}$$

Next, we will consider the stability of scattering transforms to small deformations. For $\tau \in C^1(\mathbb{R}^d; \mathbb{R}^d)$, define $T_\tau f(x) \coloneqq f(x - \tau(x))$. Note that if $\|\nabla \tau\|_{L^\infty} < 1$, then $x - \tau(x)$ is a diffeomorphism of $\mathbb{R}^d$ to itself.

Theorem 2.2.15. (Theorem 2 in [121], and Theorem 4.3 Part (d) in [44]) *Let* $0 \leq \varepsilon < 1$ *and* $R > 0$. *There exists a universal constant* $C > 0$ *(independent of* $\mathcal{G}$*) such that for all* $f \in L^2(\mathbb{R}^d)$ *which are* (ε, R)*-bandlimited, and for all* $\tau \in C^1(\mathbb{R}^d; \mathbb{R}^d)$ *with* $\|\tau\|_{L^\infty} < \infty$ *and* $\|\nabla\tau\|_{L^\infty} \leq \frac{1}{2d}$*, we have*

$$\|S[\mathcal{P}]f - S[\mathcal{P}]T_\tau f\|_{l^2L^2} \leq C(R\|\tau\|_{L^\infty} + \varepsilon)\|f\|_{L^2}. \tag{2.19}$$

Remark 2.2.16. *Note that Theorem 2 in [121], as stated, only applies to the case* $\varepsilon = 0$. *However, an analysis of the proof of Theorem 4.3 (d) in [44] (in the case where the scattering transform is an FST) reveals that the argument for the former theorem is easily extendable to the case* $\varepsilon > 0$. *Additionally, as in the case of Theorem 2.2.12, more general deformation stability results have been proven (see [32, 94, 95]), but we once again restrict ourselves to the result in [121] for ease of presentation; we leave the extension of our results to the interested reader.*

Finally, we note several properties of specific scattering transforms. Note that every scattering transform is equivariant with respect to translations. That is, if $c \in \mathbb{R}^d$, then $S[\mathcal{P}](T_c f) = T_c(S[\mathcal{P}]f)$, due to the fact that convolution and pointwise modulus commute with translation. However, it can be proven that the WST has a translation invariance property in a certain limiting sense.

Theorem 2.2.17. (Theorem 2.10 in [88]) *Let* $\mathcal{G}_{W,J}$ *be a semi-discrete wavelet frame generated by the admissible wavelet* ψ. *Then, for all* $f \in L^2(\mathbb{R}^d)$ *and all* $c \in \mathbb{R}^d$*, we have*

$$\lim_{J\to\infty} \|S[\mathcal{P}_{W,J}]f - S[\mathcal{P}_{W,J}]T_c f\|_{l^2L^2} = 0. \tag{2.20}$$

Finally, when used in practice, only finitely many layers of scattering outputs may be considered. Hence, a desirable property for scattering transforms is that higher-order layers are negligible in some

sense, and hence can be discarded as needed. This can be captured by questions of the decay rate of $\sum_{p\in\Lambda^k} \|U[p]f\|_{L^2}^2$, the energy of the k-th layer of scattering propagators, as $k \to \infty$. If this energy decays quickly, then higher-order layers will have a small fraction of the total energy of $\|S[\mathcal{P}]f\|_{l^2L^2}$ and hence can be discarded without significant effects on the other properties of the scattering transform.

In the case that the scattering transform is an FST, we can say that the aforementioned energy decays exponentially fast in k.

Theorem 2.2.18. (Proposition 3.3 in [44]) *Let $\mathcal{G}_F$ be a uniform covering frame so that $S[\mathcal{P}_F]$ is an FST. There exists a constant $C_0 \in (0,1)$ depending on $\mathcal{G}_F$ such that, for all $f \in L^2(\mathbb{R}^d)$ and $K \in \mathbb{N}$,*

$$C_0^{K-1}\|f * g_0\|_{L^2}^2 + \sum_{p\in\Lambda_F^K} \|U_F[p]f\|_{L^2}^2 \leq C_0^{K-1}\|f\|_{L^2}^2. \tag{2.21}$$

It should be noted that such a property is not unique to Fourier scattering transforms. For WSTs, exponential decay has been proven for some wavelets ψ and for a subset of square-integrable functions in the case $d = 1$ [118]. However, it has also been shown that the energy decays for WSTs can be arbitrarily slow in any dimension [52]. Additionally, exponential decay has been shown for classes of scattering transforms in [122] and [52], but in both cases exponential decay was only proved for a subset of $L^2(\mathbb{R}^d)$, rather than the whole space as in Theorem 2.2.18.

As a consequence of the exponential decay, the FST can also be shown to be norm-preserving.

Corollary 2.2.19. (Theorem 3.6 Part (a) in [44]) *If $S[\mathcal{P}_F]$ is an FST, then for all $f \in L^2(\mathbb{R}^d)$*

$$\|S[\mathcal{P}_F]f\|_{l^2L^2} = \|f\|_{L^2}. \tag{2.22}$$

A similar result can be proven for WSTs.

Theorem 2.2.20. (Theorem 2.6 in [88]) *If $S[\mathcal{P}_{W,J}]$ is a WST whose semi-discrete wavelet frame is generated by an admissible ψ, then for all $f \in L^2(\mathbb{R}^d)$*

$$\|S[\mathcal{P}_{W,J}]f\|_{l^2L^2} = \|f\|_{L^2}. \tag{2.23}$$

2.3 Strictly Stationary Processes

In many real-world settings, data is probabilistic in nature rather than deterministic. For example, many datasets are in the form of time series, which can be modeled as instances of some stochastic process. Meanwhile, many results from the study of machine learning assume that the data samples are drawn from some probabilistic distribution. Tools from machine learning, such as CNNs, have also aided in applications with such data, such as for predicting meteorological events [69, 99], forecasting gold or electricity prices [27, 84], or analyzing textures [24, 26].

A particular important class of probabilistic data is given by stationary processes, for which the statistical properties do not vary in time. Such processes are valuable for tasks like predicting future data, and a wide array of statistical mechanisms are available for both forecasting, as well as transforming non-stationary data into, stationary data [23]. In this section, we consider strictly stationary processes, in which the underlying probabilistic distributions are time-independent, and we note several properties of such processes.

2.3.1 Definitions

Let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), m)$ be our parameter space. Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ (or $\mathbb{R}^d$) be a random field. We define strict stationarity of $\{X(\omega, t) : \omega \in \Omega, t \in \mathbb{R}^d\}$

analogously to [110].

Definition 2.3.1. *A random field $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ or $\mathbb{R}^d$ is strictly stationary if, for all $k \in \mathbb{N}$, $t_1, \ldots, t_k, s \in \mathbb{R}^d$, and $A_1, \ldots, A_k \in \mathcal{B}(\mathbb{C})$ (or $\mathcal{B}(\mathbb{R}^d)$), we have:*

$$P(X(\cdot, t_1) \in A_1, \ldots, X(\cdot, t_k) \in A_k) = P(X(\cdot, t_1 + s) \in A_1, \ldots, X(\cdot, t_k + s) \in A_k).$$

Unless otherwise specified, from this point forward, we shall denote strictly stationary processes with codomain $\mathbb{C}$ with capital Latin letters, e.g., X or Y, and we will denote strictly stationary processes with codomain $\mathbb{R}^d$ with lowercase Greek letters, e.g., τ.

We will assume that X has finite second moments, i.e., $\mathbb{E}(|X(\cdot, t)|^2) < \infty$ for all $t \in \mathbb{R}^d$; note that the second moment is constant in time by strict stationarity. We also assume X is *mean square continuous*: for all $t \in \mathbb{R}^d$,

$$\lim_{s \to 0} \mathbb{E}\left[|X(\cdot, t) - X(\cdot, t + s)|^2\right] = 0. \tag{2.24}$$

For fixed $\omega \in \Omega$, we call the function $t \mapsto X(\omega, t)$ a *sample path* or *realization* of X. Given two strictly stationary processes $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ and $Y : \Omega \times \mathbb{R}^d \to \mathbb{C}$ (or $\mathbb{R}^d$), we say that they are *jointly strictly stationary* if the process $(X, Y) : \Omega \times \mathbb{R}^d \to \mathbb{C} \times \mathbb{C}$ is also strictly stationary, and similarly for $\mathbb{R}^d \times \mathbb{R}^d$-valued processes.

2.3.2 Properties

In this section, we note several important properties of strictly stationary processes which will be used extensively in Chapter 4.

2.3.2.1 Autocorrelation Function

The *autocorrelation function* $R_X : \mathbb{R}^d \to \mathbb{C}$ of a strictly stationary process X is defined by:

$$R_X(s) = \mathbb{E}\left[X(\cdot, s+t)\overline{X(\cdot, t)}\right] \text{ for all } s \in \mathbb{R}^d, \tag{2.25}$$

where the expectation is independent of t by strict stationarity of X. As X is mean square continuous, R_X is uniformly continuous [110]. Also note that $|R_X(s)| \leq R(0) = \mathbb{E}\left[|X(\cdot, t)|^2\right]$ and R_X is positive semi-definite, i.e., $\overline{R_X(s)} = R_X(-s)$ for all $s \in \mathbb{R}^d$ and, for all $n \in \mathbb{N}$, $s_1, \ldots, s_n \in \mathbb{R}^d$, and $a_1, \ldots, a_n \in \mathbb{C}$:

$$\sum_{i,j=1}^{n} a_i R_X(s_i - s_j)\overline{a_j} \geq 0.$$

We recall a special case of Bochner's Theorem:

Theorem 2.3.2. (Bochner's Theorem [111]) *Let $h : \mathbb{R}^d \to \mathbb{C}$ be continuous and positive semi-definite. Then, there exists a bounded Borel measure μ on $\mathcal{B}(\mathbb{R}^d)$ such that, for all $s \in \mathbb{R}^d$*

$$h(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} d\mu(\xi).$$

In particular, if we apply Bochner's Theorem to R_X, we see that there exists a bounded Borel measure μ_X on $\mathbb{R}^d$ such that

$$R_X(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} \mathrm{d}\mu_X(\xi). \tag{2.26}$$

2.3.2.2 Operators that Preserve Strict Stationarity

If $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$ is a Borel measure space and $g : \mathbb{C} \to \mathcal{X}$ is measurable then $g(X)$ is strictly stationary. In particular, we have that $|X|$ is strictly stationary. Similarly, if X, Y are jointly strictly stationary, then $X+Y$ is also strictly stationary. If, in addition, X, Y are both mean square continuous, then $|X|$ and $X + Y$ are also both mean square continuous.

Suppose $g \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. We define the convolution in the usual way, i.e., for $\omega \in \Omega, t \in \mathbb{R}^d$,

$$X * g(\omega, t) = \int_{\mathbb{R}^d} X(\omega, s) g(t - s) \mathrm{d}s.$$

We note several basic properties of $X * g$.

Proposition 2.3.3. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a mean square continuous, strictly stationary process, and suppose $g \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. Then, $X * g$ has the following properties:*

(1) $\mathbb{E}\left[X * g(\cdot, t)\right] = \mathbb{E}[X(\cdot, t)] \int_{\mathbb{R}^d} g(x) dx$;

(2) $\mathbb{E}\left[|X * g(\cdot, t)|\right] \leq \mathbb{E}\left[|X(\cdot, t)|\right] \|g\|_{L^1}$;

(3) $\mathbb{E}\left[|X * g(\cdot, t)|^2\right] \leq \mathbb{E}\left[|X(\cdot, t)|^2\right] \|g\|_{L^1}^2$;

(4) $\lim_{s \to 0} \mathbb{E}\left[|X * g(\cdot, t) - X * g(\cdot, t + s)|^2\right] = 0$ *for all* $t \in \mathbb{R}^d$;

(5) $R_{X*g}(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} |\widehat{g}(\xi)|^2 d\mu_X(\xi)$ *for all* $s \in \mathbb{R}^d$.

The proof of this proposition is straightforward. The proof of Part (1) in the case $d = 1$ can be found in Chapter 9.2 of [100] and is easily extendable to arbitrary d; Parts (2)-(4) are proved by a similar argument. The proof of Part (5) in the case $d = 1$ can be found in Chapter 11.9 of [48], and it is easily generalizable to our case.

Note that Part (3) of the proposition implies that $X * g$ has finite second moments and Part (4) implies that $X * g$ is mean square continuous. However, the strict stationarity of X leads to stronger results, including that $X * g$ is strictly stationary.

Proposition 2.3.4. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a mean square continuous, strictly stationary process, and suppose $g \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. Then, $X * g$ has the following properties:*

(1) $X * g$ is strictly stationary;

(2) X and $X * g$ are jointly strictly stationary;

(3) Suppose further that g is continuous and there exist constants $C, \eta > 0$ such that for all $t \in \mathbb{R}^d$

$$|g(t)| \leq \frac{C}{|t|^{d+\eta} + 1}.$$

Then, P-a.s. $X * g(\omega, t)$ is a continuous function of t.

Combining Propositions 2.3.3 and 2.3.4, we see that $X * g$ is a mean square continuous, strictly stationary process. For the reader's convenience, we provide a proof of Proposition 2.3.4 in Section 2.3.3.

2.3.3 Proof of Proposition 2.3.4

In this section, we provide a proof of Proposition 2.3.4, which details how convolution preserves strict stationarity. In order to prove these results, we first need a result on approximating strictly stationary random processes with simpler processes.

Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a strictly stationary process which is mean square continuous, and let $\delta > 0$. For $k \in \mathbb{Z}^d$, define the set $B_{k,\delta} = \prod_{i=1}^{d} (\delta(k_i - 1), \delta k_i]$. As in [110], define the *simple strictly*

stationary process $X_{\{\delta\}} : \Omega \times \mathbb{R}^d \to \mathbb{C}$ by

$$X_{\{\delta\}}(\omega, t) = \sum_{k \in \mathbb{Z}^d} X(\omega, \delta k) \mathbb{1}_{B_{k,\delta}}(t). \tag{2.27}$$

In other words, $X_{\{\delta\}}$ approximates X as $\delta \to 0$, which is formalized in Proposition 2.3.5. This sequence does so by taking a grid of cubes of side length δ, and replacing the value of $X(\omega, t)$ at any point t with the value of $X(\omega, \delta k)$, where δk is the right endpoint of the δ-cube $B_{k,\delta}$ containing t. It is easy to check that $X_{\{\delta\}}$ is also a strictly stationary random process with finite second moments and which is mean square continuous. We have the following approximation result.

Proposition 2.3.5. *Let* $\varepsilon > 0$*. Then, there exists* $\delta^* > 0$ *such that if* $0 < \delta < \delta^*$ *then*

$$\mathbb{E}\left[|X(\cdot, t) - X_{\{\delta\}}(\cdot, t)|^2\right] < \varepsilon,$$

uniformly in $t \in \mathbb{R}^d$*.*

Proof. By mean square continuity and strict stationarity, there exists $\eta > 0$ such that if $s \in \mathbb{R}^d$ with $|s| < \eta$, then

$$\mathbb{E}\left[|X(\cdot, t) - X(\cdot, t + s)|^2\right] < \varepsilon,$$

for all $t \in \mathbb{R}^d$. Clearly, there exists $\delta^* > 0$ such that $Q_{\delta^*}(0) \subseteq B_\eta(0)$. Hence, if $0 < \delta < \delta^*$ and $0 \leq s_i \leq \delta < \delta^*$ for all i, then $|s| < \eta$, where $s = (s_1, \ldots, s_d)$. If $t \in \mathbb{R}^d$, there exists a unique $k \in \mathbb{Z}^d$ such that $t \in B_{k,\delta}$. In that case, $X_{\{\delta\}}(t) = X(\delta k)$, and $0 \leq \delta k_i - t_i \leq \delta$ for all i. Thus, $|\delta k - t| < \eta$ and so

$$\mathbb{E}\left[|X(\cdot, t) - X_{\{\delta\}}(\cdot, t)|^2\right] < \varepsilon.$$

□

Now, we may move on to the proof of Proposition 2.3.4, starting with part (1). By Proposition 2.3.5, there exists a strictly decreasing sequence $\{\delta_k\}_{k=0}^{\infty}$ of positive real numbers such that $\delta_k \to 0$ and

$$\lim_{k\to\infty} \mathbb{E}\left[|X(\cdot,t) - X_{\{\delta_k\}}(\cdot,t)|^2\right] = 0,$$

uniformly in $t \in \mathbb{R}^d$. Note that

$$\mathbb{E}\left[|X * g(\cdot,t) - X_{\{\delta_k\}} * g(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X(\cdot,t) - X_{\{\delta_k\}}(\cdot,t)|^2\right] \|g\|_1^2 \to 0,$$

by Proposition 2.3.3 part (3), since clearly X and $X_{\{\delta_k\}}$ are jointly strictly stationary and hence $X - X_{\{\delta_k\}}$ is strictly stationary. In particular, $X_{\{\delta_k\}} * g(\cdot,t)$ converges in $L^2(\Omega)$ to $X * g(\cdot,t)$ for all $t \in \mathbb{R}^d$.

Recall the definition of joint distribution functions for complex-valued random variables: if $Z_1, \ldots, Z_n : \Omega \to \mathbb{C}$ are random variables, their joint distribution function $F_{Z_1,\ldots,Z_n} : \mathbb{R}^{2n} \to [0,1]$ is defined by

$$F_{Z_1,\ldots,Z_n}(a) = P\left(\text{Re}(Z_1) \leq a_1, \text{Im}(Z_1) \leq a_2, \ldots, \text{Re}(Z_n) \leq a_{2n-1}, \text{Im}(Z_n) \leq a_{2n}\right).$$

It is known from basic probability that the joint distribution functions are monotonically increasing, i.e., if $a \leq b$ (meaning $a_i \leq b_i$ for all i) then $F_{Z_1,\ldots,Z_n}(a) \leq F_{Z_1,\ldots,Z_n}(b)$. Moreover, the joint distribution function is right continuous in each variable a_i.

Assume for now that $X_{\{\delta\}} * g$ is strictly stationary for all $\delta > 0$, which we shall prove below.

Fix $t_1, \ldots, t_n, s \in \mathbb{R}^d$. Since $X_{\{\delta_k\}} * g$ converges in $L^2(\Omega)$ to $X * g$, we have that the vectors

$$(X_{\{\delta_k\}} * g(\cdot, t_1), \ldots, X_{\{\delta_k\}} * g(\cdot, t_n)) \to (X * g(\cdot, t_1), \ldots, X * g(\cdot, t_n))$$

in $L^2(\Omega)$ and hence in distribution. In particular (see Theorem 8.5 from [74]), the joint distribution functions $F_{X_{\{\delta_k\}} * g(\cdot,t_1),\ldots,X_{\{\delta_k\}} * g(\cdot,t_n)}$, $k = 1, 2, \ldots$, of $X_{\{\delta_k\}} * g(\cdot, t_i)$ converge at all continuity points $a \in \mathbb{R}^{2n}$ of $F_{X*g(\cdot,t_1),\ldots,X*g(\cdot,t_n)}$, the joint distribution function of $X * g(\cdot, t_i)$:

$$\lim_{k\to\infty} F_{X_{\{\delta_k\}} * g(\cdot,t_1),\ldots,X_{\{\delta_k\}} * g(\cdot,t_n)}(a) = F_{X*g(\cdot,t_1),\ldots,X*g(\cdot,t_n)}(a).$$

Assuming as claimed that the $X_{\{\delta_k\}} * g$ are strictly stationary, we have that

$$F_{X_{\{\delta_k\}} * g(\cdot,t_1),\ldots,X_{\{\delta_k\}} * g(\cdot,t_n)}(a) = F_{X_{\{\delta_k\}} * g(\cdot,t_1+s),\ldots,X_{\{\delta_k\}} * g(\cdot,t_n+s)}(a),$$

for all $a \in \mathbb{R}^{2n}$. Therefore, we must have that

$$F_{X*g(\cdot,t_1),\ldots,X*g(\cdot,t_n)}(a) = F_{X*g(\cdot,t_1+s),\ldots,X*g(\cdot,t_n+s)}(a),$$

for all common continuity points of the two distribution functions. By the main theorem of [75], since these two distribution functions are monotonically increasing, their discontinuity sets must have Lebesgue measure 0; thus, the set of common continuity points of the two functions must be dense in $\mathbb{R}^{2n}$. Since these two distribution functions are right continuous in each variable, it follows that the two distribution functions must be equal for all $a \in \mathbb{R}^{2n}$. In particular, this means $(X * g(\cdot, t_1), \ldots, X * g(\cdot, t_n))$ is equal in distribution to $(X * g(\cdot, t_1 + s), \ldots, X * g(\cdot, t_n + s)$. This implies that $X * g$ is

strictly stationary, as claimed.

For $M > 0$, define $f_M : \mathbb{C} \to \mathbb{C}$ by

$$f_M(z) = \begin{cases} z, & |z| \leq M, \\ M\frac{z}{|z|}, & |z| > M, \end{cases}$$

which is Lebesgue measurable. Clearly, for all $t \in \mathbb{R}^d$, $f_M(X_{\{\delta\}}) * g(\omega, t)$ converges P-a.s. to $X_{\{\delta\}} * g(\omega, t)$ as $M \to \infty$. Thus, by a similar reasoning as above (since almost sure convergence also implies convergence in distribution), if we can show that $f_M(X_{\{\delta\}}) * g$ is strictly stationary for all M, then necessarily $X_{\{\delta\}} * g$ is strictly stationary.

So, we shall assume that $|X_{\{\delta\}}| \leq M$ everywhere for some arbitrary $M > 0$. Define

$$\mathcal{X}_n = \{f : \mathbb{R}^d \to \mathbb{C} : f \text{ is constant on } B_{k',\delta/2^n}, \forall k' \in \mathbb{Z}^d, \text{ and } |f| \leq M \text{ everywhere}\}.$$

For $k_1, \ldots, k_m \in \mathbb{Z}^d$ and $A_m \in \mathcal{B}(\mathbb{C}^m)$, define the cylindrical set

$$C_{k_1,\ldots,k_m}(A_m) \coloneqq \left\{ f \in \mathcal{X}_n : \left(f|_{B_{k_1,\delta/2^n}}, \ldots, f|_{B_{k_m,\delta/2^n}} \right) \in A_m \right\}.$$

Let $\mathcal{C}_n$ be the smallest σ-algebra generated by the cylinders $C_{k_1,\ldots,k_m}(A_m)$ with $m \in \mathbb{N}$, $k_1, \ldots, k_m \in \mathbb{Z}^d$, and $A_m \in \mathcal{B}(\mathbb{C}^m)$ all arbitrary. Thus, $(\mathcal{X}_n, \mathcal{C}_n)$ is a measurable space.

For $n \in \mathbb{N}$, let $\mathcal{D}_{n,\delta} \coloneqq \{\frac{\delta k}{2^n} : k \in \mathbb{Z}^d\}$. Suppose $t' \in \mathcal{D}_{n,\delta}$. For any fixed $\omega \in \Omega$, notice that $X_{\{\delta\}}(\omega, \cdot) \in \mathcal{X}_n$. Indeed, $X_{\{\delta\}}(\omega, \cdot)$ is constant on $B_{k,\delta}$ and each $B_{k',\delta/2^n}$ is a subset of some unique $B_{k,\delta}$. Similarly, $X_{\{\delta\}}(\omega, \cdot + t')$ is constant on $B_{k,\delta} - t'$, and since $t' \in \mathcal{D}_{n,\delta}$, it must be constant on each $B_{k',\delta/2^n}$. Hence, $X_{\{\delta\}}(\omega, \cdot + t') \in \mathcal{X}_n$. It is easy to check that the maps $\omega \mapsto X_{\{\delta\}}(\omega, \cdot)$ and

$\omega \mapsto X_{\{\delta\}}(\omega, \cdot + t')$ are measurable maps from Ω to $\mathcal{X}_n$.

Now, let $C_{k_1,\ldots,k_m}(A)$ be a cylinder. Using strict stationarity, we see that

$$\begin{aligned}
&P\left(\omega : X_{\{\delta\}}(\omega, \cdot) \in C_{k_1,\ldots,k_m}(A)\right) \\
&= P\left(\omega : \left(X_{\{\delta\}}\left(\omega, \frac{\delta k_1}{2^n}\right), \ldots, X_{\{\delta\}}\left(\omega, \frac{\delta k_m}{2^n}\right)\right) \in A\right) \\
&= P\left(\omega : \left(X_{\{\delta\}}\left(\omega, \frac{\delta k_1}{2^n} + t'\right), \ldots, X_{\{\delta\}}\left(\omega, \frac{\delta k_m}{2^n} + t'\right)\right) \in A\right) \\
&= P\left(\omega : X_{\{\delta\}}(\omega, \cdot + t') \in C_{k_1,\ldots,k_m}(A)\right).
\end{aligned}$$

Hence, $X_{\{\delta\}}(\omega, \cdot)$ and $X_{\{\delta\}}(\omega, \cdot + t')$ are equal in distribution on the finite cylinders, and hence must be equal in distribution on all of $\mathcal{C}_n$. That is, for all $C' \in \mathcal{C}_n$, we have

$$P\left(\omega : X_{\{\delta\}}(\omega, \cdot) \in C'\right) = P\left(\omega : X_{\{\delta\}}(\omega, \cdot + t') \in C'\right).$$

Next, for $t_1, \ldots, t_l \in \mathbb{R}^d$ and $A_l \in \mathcal{B}(\mathbb{C}^l)$, consider the cylindrical set

$$\widetilde{C}_{t_1,\ldots,t_l}(A_l) \coloneqq \left\{h : \mathbb{R}^d \to \mathbb{C} : (h(t_1), \ldots, h(t_l)) \in A_l\right\}.$$

Let $\widetilde{\mathcal{C}}$ be the smallest σ-algebra generated by all such cylinders for $l \in \mathbb{N}, t_1, \ldots, t_l \in \mathbb{R}^d$ and $A_l \in \mathcal{B}(\mathbb{C}^l)$ arbitrary. Then, $(\{h : \mathbb{R}^d \to \mathbb{C}\}, \widetilde{\mathcal{C}})$ is a measurable space.

Define $\widetilde{\Psi} : \mathcal{X}_n \to \{h : \mathbb{R}^d \to \mathbb{C}\}$ by

$$\widetilde{\Psi} f(t) \coloneqq \int_{\mathbb{R}^d} f(s) g(t - s) \mathrm{d}s. \tag{2.28}$$

That is, $\widetilde{\Psi}$ is a convolution with g. Since $f \in \mathcal{X}_n$ satisfies $|f| \leq M$ everywhere and $g \in L^1(\mathbb{R}^d)$, this map is well-defined. We claim this map is measurable with respect to the σ-algebras $\mathcal{C}_n$ and $\widetilde{\mathcal{C}}$. It suffices to show that all of the coordinates are measurable, i.e., that $\widetilde{\Psi}_t : \mathcal{X}_n \to \mathbb{C}$ defined by $\widetilde{\Psi}_t f = \widetilde{\Psi} f(t)$ is measurable for all $t \in \mathbb{R}^d$. Indeed, if this is true, then it is easy to show that $\widetilde{\Psi}^{-1}(\widetilde{C}_{t_1,\ldots,t_l}(A)_l) \in \mathcal{C}_n$ for every cylinder $\widetilde{C}_{t_1,\ldots,t_l}(A_l)$ as $\widetilde{\Psi}_{t_1}, \ldots, \widetilde{\Psi}_{t_l}$ are measurable. Thus, since $\{\widetilde{C}_{t_1,\ldots,t_l}(A_l)\}$ generate $\widetilde{\mathcal{C}}$, we have that $\widetilde{\Psi}$ is measurable.

Fix $t \in \mathbb{R}^d$. If $f \in \mathcal{X}_n$, then f is piecewise-constant, and we can write

$$f(s) = \sum_{k \in \mathbb{Z}^d} f\left(\frac{\delta k}{2^n}\right) \mathbb{1}_{B_{k,\delta/2^n}}(s).$$

This implies that

$$\widetilde{\Psi}_t f = \sum_{k \in \mathbb{Z}^d} f\left(\frac{\delta k}{2^n}\right) \int_{B_{k,\delta/2^n}} g(t-s)\mathrm{d}s.$$

For each $k \in \mathbb{Z}^d$, $\int\limits_{B_{k,\delta/2^n}} g(t-s)\mathrm{d}s$ is independent of f and is a function of $t \in \mathbb{R}^d$. Thus, the partial sums of $\widetilde{\Psi}_t$ are clearly measurable (just being linear combinations of the values $f|_{B_{k,\delta/2^n}}$) and thus $\widetilde{\Psi}_t$ must be measurable. Since t was arbitrary, $\widetilde{\Psi}$ is measurable.

Consider the map $\mathscr{G} : (\Omega, \mathcal{F}) \to (\{h : \mathbb{R}^d \to \mathbb{C}\}, \widetilde{\mathcal{C}})$ given by

$$\omega \overset{\mathscr{G}}{\mapsto} \left(t \mapsto \widetilde{\Psi}(X_{\{\delta\}}(\omega, \cdot))(t)\right) = \omega \overset{\mathscr{G}}{\mapsto} \left((X_{\{\delta\}}(\omega, \cdot) * g)(t)\right),$$

which is measurable as it is the composition of measurable maps. For a cylindrical set $\widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A}) \in \widetilde{C}$

where $\widetilde{A} = A_1 \times \cdots \times A_l$ for $A_j \in \mathcal{B}(\mathbb{C})$, notice that

$$\begin{aligned} &P\big(\omega : \mathscr{G}(\omega) \in \widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A})\big) = \ P\big(\omega : \widetilde{\Psi}(X_{\{\delta\}}(\omega,\cdot)) \in \widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A})\big) \\ &= P\big(\omega : X_{\{\delta\}}(\omega,\cdot) \in \widetilde{\Psi}^{-1}(\widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A}))\big) = \ P\big(\omega : X_{\{\delta\}}(\omega,\cdot+t') \in \widetilde{\Psi}^{-1}(\widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A}))\big) \\ &= P\big(\omega : \widetilde{\Psi}(X_{\{\delta\}}(\omega,\cdot+t')) \in \widetilde{C}_{t_1,\ldots,t_l}(\widetilde{A})\big). \end{aligned}$$

Notice that $\widetilde{\Psi}(X_{\{\delta\}}(\omega,\cdot))(t) = X_\delta(\omega,\cdot) * g(t)$. Also, notice that $\widetilde{\Psi}(X_{\{\delta\}}(\omega,\cdot+t'))(t) = X_{\{\delta\}}(\omega,\cdot+t') * g(t) = X_{\{\delta\}}(\omega,\cdot) * g(t+t')$. Thus, we see that

$$\begin{aligned} &P\big(X_{\{\delta\}} * g(\cdot,t_1) \in A_1,\ldots,X_{\{\delta\}} * g(\cdot,t_l) \in A_l\big) \\ &= P\big(X_{\{\delta\}} * g(\cdot,t_1+t') \in A_1,\ldots,X_{\{\delta\}} * g(\cdot,t_l+t') \in A_l\big). \end{aligned}$$

Hence, for any $n \in \mathbb{N}$ and any $t' \in \mathcal{D}_{n,\delta}$, the processes $X_{\{\delta\}} * g(\omega,t)$ and $X_{\{\delta\}} * g(\omega,t+t')$ have the same finite-dimensional distributions.

Notice that $\mathcal{D}_\delta \coloneqq \bigcup_{n=1}^{\infty} \mathcal{D}_{n,\delta}$ is dense in $\mathbb{R}^d$. Also, by Proposition 2.3.3 part (4) we have that

$$\lim_{r\to 0} \mathbb{E}\left[|X_{\{\delta\}} * g(\cdot,t) - X_{\{\delta\}} * g(\cdot,t+r)|^2\right] = 0,$$

uniformly in $t \in \mathbb{R}^d$. Thus, for $t' \in \mathbb{R}^d$, we can choose a sequence $\{(t')_m\} \subseteq \mathcal{D}_\delta$ that converges to t', in which case $X_{\{\delta\}} * g(\cdot,t+(t')_m)$ converges in $L^2(\Omega)$ to $X_{\{\delta\}} * g(\cdot,t+t')$. Arguing as before, since $(X_{\{\delta\}} * g(\cdot,t_1),\ldots,X_{\{\delta\}} * g(\cdot,t_l))$ is equal in distribution to $(X_{\{\delta\}} * g(\cdot,t_1+(t')_m),\ldots,X_{\{\delta\}} * g(\cdot,t_l+(t')_m))$ for fixed $t_1,\ldots,t_l$, l arbitrary, and for all m, we must have that $(X_{\{\delta\}} * g(\cdot,t_1),\ldots,X_{\{\delta\}} * g(\cdot,t_l))$ is equal in distribution to $(X_{\{\delta\}} * g(\cdot,t_1+t'),\ldots,X_{\{\delta\}} * g(\cdot,t_l+t'))$. Thus, $X_{\{\delta\}} * g$ is strictly stationary,

and so $X * g$ is strictly stationary. This proves part (1).

Following a similar argument, it can also be shown that X and $X*g$ are jointly strictly stationary, thus proving part (2).

It remains to prove part (3). So, suppose that g is continuous and that there exist constants $C, \eta > 0$ such that $|g(t)| \leq \frac{C}{|t|^{d+\eta}+1}$ for all $t \in \mathbb{R}^d$. Define $h : \mathbb{R}^d \to \mathbb{C}$ by $h(t) \coloneqq \sup_{|r|\leq 1} |g(t-r)|$. Clearly, for any $|r| \leq 1$, we have that $|g(t-r)| \leq h(t)$. In particular, $|g(t)| \leq h(t)$. Next, notice that h is uniformly bounded by C. Moreover, if $|t| > 1$, then for all $|r| \leq 1$ we have $|t-r| \geq |t|-|r| \geq |t|-1$. Thus, for $|r| \leq 1$, we see that

$$|g(t-r)| \leq \frac{C}{|t-r|^{d+\eta}+1} \leq \frac{C}{(|t|-1)^{d+\eta}+1}.$$

In particular, this means that $|h(t)| \leq \frac{C}{(|t|-1)^{d+\eta}+1}$ for all $|t| > 1$. Since the right hand side is clearly integrable and square-integrable on $\mathbb{R}^d \setminus B_1(0)$, we see that $h \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$.

As a consequence of Proposition 2.3.3 part (1) and the Fubini-Tonelli Theorem, for each $t \in \mathbb{R}^d$, $X * g(\omega, t)$ exists P-a.s. By the same reasoning, we also get that for each $t \in R^d$, $X * h(\omega, t)$ exists P-a.s. Let $T \subseteq \mathbb{R}^d$ be a dense, countable subset. By σ-additivity, there thus exists a set $F \subseteq \Omega$ such that $P(F) = 1$ and $X * g(\omega, t), X * h(\omega, t)$ exist for all $\omega \in F, t \in T$.

Now, let $t' \in \mathbb{R}^d$ be arbitrary. By density, there exists a sequence $\{t_j\}_{j=1}^{\infty} \subseteq T$ such that $|t_j - t'| \leq \frac{1}{2}$ for all j and $|t_j - t'| \to 0$. Fix $\omega \in F$. For all $s \in \mathbb{R}^d$, notice that $X(\omega, s)g(t_j - s) \to X(\omega, s)g(t' - s)$ since g is continuous.

Next, choose $t'' \in T$ such that $|t'' - t'| < \frac{1}{2}$. Then, notice that $|(t_j - s) - (t'' - s)| = |t_j - t''| \leq |t_j - t'| + |t' - t''| \leq 1$ for all j, s. Thus, we have that $|g(t_j - s)| \leq \sup_{|r|\leq 1} |g((t'' - s) - r)| = h(t'' - s)$ for all j. Since $t'' \in T$, we have that $X * h(\omega, t'')$ exists, i.e., $X(\omega, s)h(t'' - s)$ is integrable in s. Since

$|X(\omega,s)g(t_j-s)| \leq |X(\omega,s)h(t''-s)|$, by the Lebesgue dominated convergence theorem we get that $X(\omega,s)g(t'-s)$ is integrable in s, i.e., $X*g(\omega,t')$ exists. Moreover, we get that $X*g(\omega,t_j) \to X*g(\omega,t')$, and so it is easily seen that $X*g(\omega,\cdot)$ is continuous at t' for each $\omega \in F$.

Since t' was arbitrary, we see that $X*g(\omega,t)$ exists for all $\omega \in F$ and all $t \in \mathbb{R}^d$, and is a continuous function in t for fixed ω. That is, $X*g$ P-a.s. has continuous realizations, proving Part (3). This proves Proposition 2.3.4.

2.4 Inverting the Discretized Wavelet Scattering Transform

Neural networks have seen immense success in the problem of generating new instances of a given data distribution. Particularly notable examples of such networks come in the form of Generative Adversarial Networks (GANs), originally introduced in 2014 [57]. A GAN consists of two separate neural networks: a generator $\mathfrak{G}$, which turns a sample from an a priori distribution (e.g., a multivariate Gaussian) into an element of the data space, and a discriminator $\mathfrak{D}$, which attempts to tell apart real and generated data. These networks are trained in opposition to each other through a minimax game, and they are trained until the discriminator can't distinguish the generated data from the real data.

Several different network architectures have been proposed for the two GAN networks. In the original paper, $\mathfrak{G}$ and $\mathfrak{D}$ were both standard feed forward networks [57], while [107] achieved superior results by using CNNs. However, training two separate networks requires significant training time, and extreme care must be taken during training to prevent model collapse and overfitting of the discriminator [57].

Thus, Mallat and Tomás Angles proposed a new method for generation problems [14]. While they still use a generator network to map from the fixed probability distribution to the data distribution,

the discriminator is replaced with a wavelet scattering transform-based operation that maps the data distribution back to the fixed distribution. This not only provides a means for generating novel data in the data space, but it also provides an approximate inverse for a discretized WST.

2.4.1 Wavelet Scattering Transform in Applications

Before we introduce Mallat's and Angles's method for generating data, we shall first explicitly describe the discretized wavelet scattering transform. While various implementations of the WST have been created, we focus on the implementation in the package Kymatio [13]. Moreover, as we will only be considering datasets consisting of images in this dissertation, we will focus on the implementation for two-dimensional data.

Let $\psi : \mathbb{R}^2 \to \mathbb{C}$ be a mother wavelet. Since we can only apply finitely many convolution filters, fix $J, Q \in \mathbb{N}$, where J is the number of scales (or dilations) of ψ and and Q is the number of used directions (or rotations). For integers $1 \leq j \leq J$ and $1 \leq q \leq Q$, we define $\psi_{j,q}(x) \coloneqq 2^{-2j}\psi(2^{-j}r_{\theta_q}x)$, where $\theta_q = \frac{\pi q}{Q}$ and r_{θ_q} is the rotation (in the plane) by angle θ_q. Also, let $\phi : \mathbb{R}^2 \to \mathbb{C}$ be a low-pass filter, and define $\phi_J(x) = 2^{-2J}\phi(2^{-J}x)$.

In this implementation, ψ is a Morlet wavelet:

$$\psi(x) = C_1 e^{\frac{-|x|^2}{2\sigma^2}}(e^{i\xi\cdot x} - C_2) \tag{2.29}$$

where $\sigma > 0$ is the standard deviation of the wave packet, $\xi \in \mathbb{R}^2$ is a constant direction, C_1 is a normalizing constant, and C_2 is a constant chosen so that $\int \psi(x)\mathrm{d}x = 0$. In addition, ϕ is a Gaussian:

$$\phi(x) = C_3 e^{\frac{-|x|^2}{2\sigma^2}}. \tag{2.30}$$

More specifically, in [13] C_2 and C_3 are chosen so that $\int |\psi(x)|\mathrm{d}x = \int |\phi(x)|\mathrm{d}x = 1$, while $\sigma = 0.8$ and $\xi = \begin{bmatrix} \frac{3\pi}{4} & 0 \end{bmatrix}^T$. Note that these choices of ψ and ϕ, in the continuous setting, do not generate a semi-discrete wavelet frame as defined by equation 2.14, but they do satisfy, for all $\xi \in \mathbb{R}^d$,

$$(1-\varepsilon) \leq |\widehat{\phi_J}(\xi)|^2 + \sum_{\lambda \in \Lambda_{W,J}} |\widehat{\psi_\lambda}(\xi)|^2 \leq 1 \tag{2.31}$$

for some fixed $0 < \varepsilon < 1$ (in [26], if $\sigma = 0.7$ then $\varepsilon = 0.25$). Both of these functions are appropriately discretized for application (with all integrals being replaced by sums).

Assume that F is now an image, instead of a function on $\mathbb{R}^2$. That is, F is a function from $\{1, \ldots, M\} \times \{1, \ldots, N\}$ to $\mathbb{R}$, where M and N are the height and width of the image, respectively; F can also be represented as an $M \times N$ matrix. We then define the (2-layer) wavelet scattering transform of F, $S[J, Q]F$, as the union of the following three sets:

$$\{F \star \phi_J\} \tag{2.32}$$

$$\{|F \star \psi_{j,q}| \star \phi_J\}_{\substack{1 \leq j \leq J \\ 1 \leq q \leq Q}} \tag{2.33}$$

$$\{||F \star \psi_{j,q}| \star \psi_{j',q'}| \star \phi_J\}_{\substack{1 \leq j < j' \leq J \\ 1 \leq q, q' \leq Q}} \tag{2.34}$$

where $\star$ denotes (periodic) convolution. Note that each element of one these sets is itself a 2 - dimensional image; we shall therefore refer to each one of these images as a *scattering channel*. We will call each pixel of a scattering channel a *scattering coefficient*, and we call each of the above sets a *layer*.

There are several things to note with this definition. First, while it is known in the continuous setting that layer energies of the WST can decay arbitrarily slowly [52], for practical settings like

images, higher-order layers have been observed to produce negligible coefficients. Thus, only the first two layers of scattering coefficients are calculated. A similar phenomenon has been observed for $1 \leq j' \leq j$ [26], and thus only filters with larger j' are considered in the second layer.. Also, since F and $\widehat{\psi}$ (being a difference of two Gaussians) are real-valued, we have by Remark 2.2.6 that we only need to calculate the scattering outputs for angles in $(0, \pi]$, which is why we only consider the angles $\frac{\pi q}{Q}$ for $1 \leq q \leq Q$.

Finally, as a high number of different outputs are produced by $S[J, Q]F$, this scattering transform representation is known to exhibit redundancy. Hence, in this implementation each scattering channel is downsampled by a factor of 2^J in both the width and height. Thus, if the input is of dimension (M, N), then the scattering transform output will be of dimension $(S, \frac{M}{2^J}, \frac{N}{2^J})$, where $S = 1 + JQ + \frac{J(J-1)Q^2}{2}$ is the number of scattering channels.

We can also extend the definition $S[J, Q]$ to 2D images which consist of multiple matrices (or *channels*) representing different information (see Section 2.5 for examples of such images). So, let F be a function from $\{1, \ldots, \widehat{C}\} \times \{1, \ldots, M\} \times \{1, \ldots, N\}$ to $\mathbb{R}$, i.e., a $\widehat{C} \times M \times N$ matrix, where $\widehat{C}$ is the number of channels and M and N are the dimensions of each channel (we use the notation $\widehat{C}$ because this term represents the spectral dimension). For $c \in \{1, \ldots, \widehat{C}\}$, let $F_c \coloneqq F(c, \cdot, \cdot)$ be the c-th channel. Then, we compute $S[J, Q]F$ by computing $S[J, Q]F_c$ for all c and then combining each output into a full tensor; that is, the output of $S[J, Q]F$ is a tensor of dimension $(\widehat{C}, S, \frac{M}{2^J}, \frac{N}{2^J})$.

2.4.2 Generation and Inversion

Now, we detail the two components of Mallat and Angles's scattering transform-based method for image generation. The first component embeds the images into a multivariate Gaussian distribution.

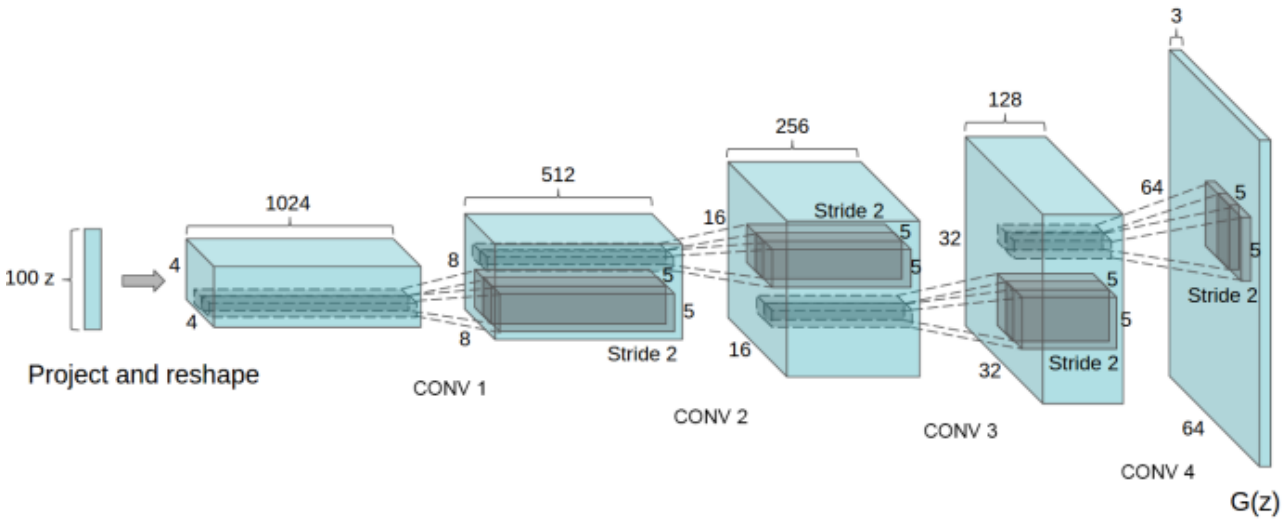


Figure 2.1: (Figure 1 from [107]) Example architecture of the generator of a DCGAN. In this example a 100-dimensional Gaussian distribution is mapped to an image of size $3 \times 64 \times 64$.

Given an image F (with or without channels), we compute the scattering transform $S[J, Q]F$. Note that, as explained in [14], increasing J increases the amount of Gaussianization of the data. However, it also increasingly contracts distances between distinct elements of the data set (i.e., reduces the bi-Lipschitzness), leading to an inherent tradeoff. After the scattering transform is calculated, an affine operator is applied to the output to whiten the data and project it a smaller, d-dimensional space (e.g., through principal component analysis (PCA) [65, 101]).

The second component maps the d-dimensional space back to the image data. It does so by using a trained neural network which has the same architecture as the generator of a Deep Convolutional GAN (DCGAN) [107]; see Figure 2.1. It first applies a fully connected layer (with bias) followed by batch normalization and ReLU activation function, where $\text{ReLU}(x) = \max\{0, x\}$. This layer maps the d-dimensional embedding into a tensor of size $(1024, \frac{M}{2^J}, \frac{N}{2^J})$.

Next, J convolutional blocks are applied. Each block starts by upsampling the spatial dimen-

sions by a factor of 2. It does so using bilinear interpolation [106]: if a function g is known at the values (x_i, y_j) for all $i, j \in \{1, 2\}$, then for $x_1 < x < x_2$ and $y_1 < y < y_2$ we interpolate the value $g(x, y)$ by

$$g(x,y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) \\ f(x_2, y_1) & f(x_2, y_2) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}. \tag{2.35}$$

Then, a convolutional layer without bias is applied, with the number of applied filters equal to half of the number of channels of the input data to this layer. Each of these blocks ends with batch normalization and ReLU activation. Finally, one more convolutional block is applied, consisting of a convolutional layer (with the number of filters equal to the number of channels of the images in the dataset), followed by batch normalization and $\tanh$ activation, where $\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

The generator network is trained to invert the embedding of the input images into the d - dimensional space. It is trained with l^1 loss, as it had been shown to give improved results when generating images [22]. They tested their method on 3 different image datasets: Polygon5, a collection of random colored, convex polygons with at most 5 vertices; CelebA [83], a collection of faces of celebrities; and LSUN [126], a collection of images of bedrooms. All images in these datasets are of size $3 \times 128 \times 128$, and they used a WST with $J = 4$ scales and $Q = 8$ rotations. Examples of the outputs of the inverse network for the test set can be seen in Figure 2.2.

2.5 Hyperspectral Images and Reconstruction

The modern world is becoming dominated by high-dimensional, noisy, complex data, where often the same phenomena can be described by different types, or *modalities*, of data from different

Figure 2.2: (Figure 3 from [14]) Example reconstructions after applying the discretized WST and whitening operator, followed by the DCGAN network, for images in the test sets of Polygon5, CelebA, and LSUN. The top images are the original images, and the bottom images are the reconstructions.

sensors or experiments. A particularly notable example is given by hyperspectral imaging (HSI), which produces images which contains the spectral information of the subject as opposed to standard color, or RGB, images, which only contain the red, green, and blue color information. However, HSI are difficult and expensive to obtain, leading to a desire for algorithms that will generate HSI from easier-to-obtain images. In this section, we discuss what HSI are and how they are obtained, the hyperspectral reconstruction problem, and two datasets for hyperspectral reconstruction.

2.5.1 Hyperspectral Images

Standard color images, i.e., RGB images, are represented using 3 separate matrices which correspond to the red, green, and blue colors of the image subject; each matrix is called a *channel*. So, for an RGB image with height M and width N, the image is stored as a $3 \times M \times N$ tensor, where 3 is the number of channels. A hyperspectral image (HSI), however, consists of many different channels, with each one being measured from a different wavelength of light. These wavelengths span a section of the electromagnetic (EM) spectra, from the visual (VIS) range ($\sim 400-700$ nm) through the near-infrared (NIR) range ($\sim 700-1400$ nm) to the short-wavelength infrared (SWI) range ($\sim 1400-2500$ nm) [21]. Like RGB images, HSI are stored as $\widehat{C} \times M \times N$ tensors, where $\widehat{C}$ is the number of channels. A related image modality is multispectral images (MSI), which are HSI of size $\widehat{c} \times M \times N$ where the number of channels $3 \leq \widehat{c} \ll \widehat{C}$.

HSI are obtained through the use of various sensors and cameras. Sensors are used to collect the EM radiation that is reflected by the subject of the image. Then, an imaging spectrometer divides the data into the separate spectral bands [64]. These devices may be attached to satellites, planes, or drones for acquiring these images, or they may be found in specialized hyperspectral cameras [21].

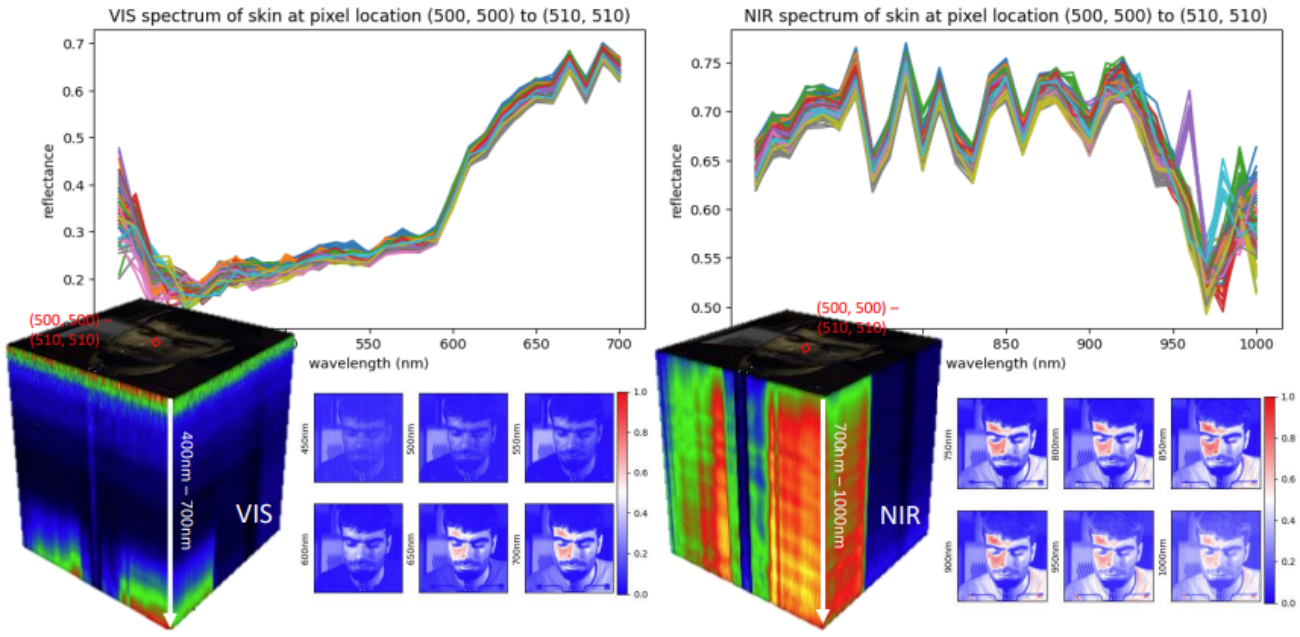


Figure 2.3: (Figure 1 from [92]) An example HSI from the dataset Hyper-Skin. On the left, we have the channels from the visual (VIS) range, and on the right, we have the channels from the near-infrared (NIR) range. The graphs are of the spectra from the spatial center of the image.

One can think of a $\widehat{C} \times M \times N$ HSI as an $M \times N$ image, where each pixel is in actuality a vector in $\mathbb{R}^{\widehat{C}}$, called a *spectrum*. These spectra measure the spectral response of the material(s) in that pixel. As different materials have different EM responses, HSI thus encode information that can be used to more easily distinguish between different objects and materials in computer vision tasks. HSI have thus seen use in a wide variety of fields and problems, including food safety inspections, precision agriculture, medical diagnoses, authentication of forensic documents and artwork, landmine and anomaly detection, and flood detection and management [21, 30, 64, 73, 112]. An example HSI is shown in Figure 2.3.

2.5.2 Hyperspectral Reconstruction

While HSI have proven to be extremely valuable image modalities for a wide variety of applications, their widespread use has been hampered by several factors [21]. First, the equipment used for capturing HSI are quite expensive. Moreover, the cameras are highly sensitive to user error, and light reflecting from the ambient surroundings can also cause errors in the measurements of the spectra. Additionally, when obtaining the images, there is often a tradeoff between the spatial and spectral resolution: an increase in the number of pixels typically requires that fewer channel measurements are collected, and vice versa.

Due to these factors, there has been a rise of interest and study of the problem of *hyperspectral reconstruction*. The goal of hyperspectral reconstruction, or *spectral superresolution*, is to generate HSI from images which are smaller and easier to obtain. Algorithms for this task take as input RGB images or MSI with only a few spectral bands, and they proceed to reconstruct the missing spectral bands of an HSI of the same subject as the input. As RGB images and MSI can be significantly easier to obtain, hyperspectral reconstruction significantly reduces the financial and human costs of obtaining HSI. Resources can also then be allocated towards capturing a higher spatial resolution for the input images, which, after hyperspectral reconstruction, can lead to HSI with both high spatial and spectral resolutions.

The hyperspectral reconstruction problem has seen numerous proposed solutions. Several prior-based methods which use known information about the HSI to be reconstructed have been advanced. These include dictionary-based methods and sparse representations [1, 15, 109], as well as manifold-learning techniques based on the knowledge that the HSI lie on low-dimensional manifolds [67]. Analysis in terms of a Gaussian process has also been used to approximate the relatively smooth spectra of

natural images [2] . However, these methods all suffer from a lack of generalizability beyond specific cases of HSI [129].

Numerous deep learning approaches have also been studied for the hyperspectral reconstruction problem, including a variety of deep CNNs for reconstruction [72, 113, 117, 123, 124, 130]. Generative adversarial networks (GANs) [77, 82, 132] and residual networks [81, 133] have also been used to improve image quality and details. More complex architectures with a high number of layers and parameters have been explored more recently, such as networks which use attention blocks [66,68,77,103,127] and transformer-based recurrent neural networks [28,29,78,134] (we will discuss the model from [29] in more detail in Section 2.5.3.3). Fusion networks, involving several of the above techniques, have also been explored [47, 135]. However, all of these models have complex architectures and a large number of parameters, which require both considerable computational resources and substantial training time.

2.5.3 Datasets and Baseline for Hyperspectral Reconstruction

Hyperspectral reconstruction has also been the subject of multiple grand challenges at several different research conferences. For example, at the 2022 New Trends in Image Restoration and Enhancement (NTIRE) workshop, in conjunction with the Conference on Computer Vision and Pattern Recognition (CVPR), a reconstruction challenge was held in which participants had to reconstruct HSI from the ARAD dataset from corresponding RGB images [16]. Similarly, the Grand Challenge on Hyperspectral Skin Vision was held as part of the 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), in which participants had to reconstruct HSI from the Hyper-Skin dataset from corresponding MSI [93]. In this section, we describe these two datasets as

well as a baseline hyperspectral reconstruction method.

2.5.3.1 ARAD Dataset

The ARAD dataset was used for the 2022 NTIRE hyperspectral reconstruction challenge [16]. This dataset consists of 1000 pairs of images, an RGB image and an HSI, of a wide variety of subjects and scenes, including parks, murals, statues, flowers, buildings, and beaches. The image pairs in the dataset were labeled 1 through 1000, and they were divided into a training set (images 1–900), a validation set (images 901–950), and a testing set (images 951–1000). The full training and validation sets were provided to the participants of the challenge, while only the RGB images of the testing set were provided in order to allow for fair evaluation by the challenge organizers. Examples of the RGB images can be found in Figure 2.4.

The HSI were collected by a Specim IQ mobile hyperspectral camera. This camera initially produces HSI with 204 channels, each of size 512×512, where the channels were measured from wavelengths in the range 400–1000 nm. For the challenge, these images were resampled in the spectral dimension to produce 31 channels in the visual range, i.e., 400–700 nm, with 10 nm increments. Additionally, columns with excessive interference from the ambient surroundings were removed, resulting in final HSI of size $31 \times 482 \times 512$. The RGB images were then created from the HSI by the challenge organizers using a pipeline made to simulate a camera in a realistic real-world setting. The color channels were created using the camera's spectral response function, and a noise model was applied and the images were compressed using the JPEG format.

While the original training and validation sets are available to the public, the test HSI have never been released. Thus, for our own experiments (see Chapters 5 and 6), we needed to create our own

Figure 2.4: Example RGB images from the ARAD dataset. All images are from the training set.

testing set. To do so, we removed every 18th element of the training set, starting at image 18, and made them into our test set. Since the original training set had 900 images, our new training set has 850 images, while our validation and testing sets each have 50 images. Additionally, we chose to 0-pad all of the RGB images and HSI to make square channels of size 512×512, as some of our techniques (such as downsampling and upsampling) are more robust when the sizes of our channels are powers of 2.

2.5.3.2 Hyper-Skin Dataset

The Hyper-Skin dataset [92] was used as part of the 2024 ICASSP Grand Challenge on Hyperspectral Skin Vision [93]. The dataset consists of pairs of images, one MSI and one HSI, of the faces and skin of 51 different participants. For each participant, 6 images were obtained, with 3 different head positions (facing forward, to the left, or to the right) and 2 different facial expressions (neutral or smiling). The images from 7 of these participants (namely, participant numbers 1, 2, 16, 30, 35, 36, and 39) were removed from the set as part of the testing set; the images from the remaining 44 participants formed the training set for the competition, for a set of size 264 image pairs. Examples of the MSI can be found in Figure 2.5.

A Specim FX10 camera, which was moved on a custom scanner, was used to acquire the HSI. Initially, 448 spectral bands were measured, spanning the range from 400 to 1000 nm, and all of spatial size 1024×1024. The challenge organizers then used SciPy's interpolation function to downsample the number of channels to 61, from 400 to 1000 nm in 10 nm increments. The channels are divided into two sections: the visual (VIS) range, from 400 to 700 nm, and the near-infrared (NIR) range, from 700 to 1000 nm.

Figure 2.5: Example MSI from the Hyper-Skin dataset. The color images are RGB, and the grayscale images are the corresponding 960 nm channel. The left two columns are of participant 3, and the right two columns are participant 2.

The challenge organizers then created RGB images using the HSI2RGB simulation pipeline, which incorporated 28 different camera response functions [87]. They then concatenated the RGB images with the 960 nm channel from the HSI to create the MSI which are used as the inputs for this hyperspectral reconstruction challenge. Additional MSI were created for participants 3 and 12 but with different generated cameras, such as those found in smartphones, in order to test the generalizability of the proposed models to different cameras. These additional images were added to the testing set, for a total of 54 images from 9 participants.

As in the case of the ARAD dataset, the testing set HSI were not made available to the public. Additionally, the Hyper-Skin dataset did not have a predefined validation set, requiring us to construct our own validation and testing sets for our experiments. For our validation set, we removed all images from participants 7, 27, 29, and 41; for the testing set, we removed all images from participants 3, 17, 33, and 50 (all of these participants were chosen at random). Thus, our training set has 36 participants with 6 images each, for a total of 216 images, while the validation and testing sets each have 24 images.

2.5.3.3 Baseline Network

The organizers of the Hyper-Skin challenge used the Multi-stage Spectral Wise Transformer (MST++) model as a baseline for comparisons [29]. MST++ is a transformer-based architecture designed for hyperspectral reconstruction, and it was the top performing model for the NTIRE hyperspectral reconstruction challenge on ARAD. This model mainly consists of repeated encoders, bottlenecks, and decoders, which are individually made up of various convolutional and feed forward layers, skip connections, batch normalizations, and attention blocks. A diagram of MST++'s architecture can be seen in Figure 2.6. In the diagram, $\times N_i$ means that a block of the model is applied

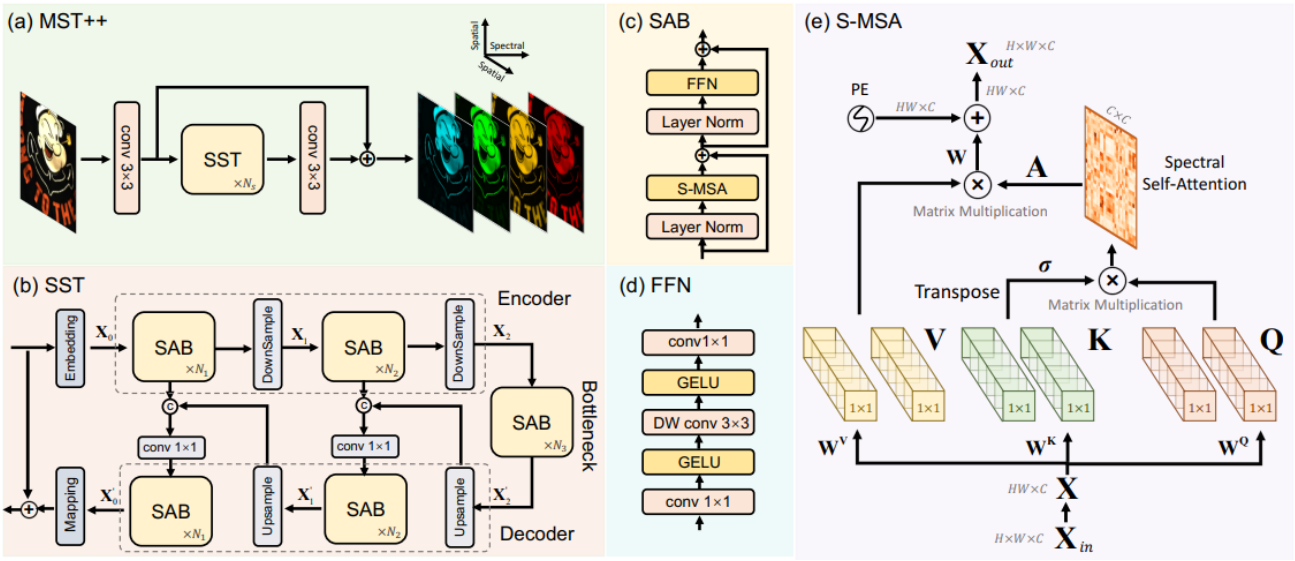


Figure 2.6: (Figure 2 from [29]) A diagram of the model architecture of MST++.

N_i times, and $GELU$ is a Gaussian Error Linear Unit activation function [61] , which is defined by $\text{GELU}(x) = x\Phi(x)$, where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x = \frac{x}{2}\left[1 + \text{erf}(x/\sqrt{2})\right]$ is the distribution of a standard Gaussian.

For the baseline comparison, the organizers of the Grand Challenge trained an implementation of MST++ with $N_1 = N_2 = N_3 = 1$ and $N_S = 3$ on the Hyper-Skin dataset. Due to memory constraints, the model was trained to reconstruct 128×128 patches of the HSI (from the corresponding MSI patches) and then tested on the full size data. MST++ was trained for 100 epochs using the l^1 loss function, with a batch size of 2. The Adam optimizer was used with an initial learning rate of 4×10^{-4} and $(\beta_1, \beta_2) = (0.9, 0.999)$ (which control the computation of the gradients and their squares); a cosine annealing scheduler was also used.

Chapter 3: Limiting Translation Invariant Behavior of Scattering Transforms

3.1 Introduction

Many properties of wavelet scattering transforms, such as boundedness (Corollary 2.2.14), Lipschitzity (Theorem 2.2.12), and stability to small deformations (Theorem 2.2.15), have been proven for more general constructions of scattering transforms. However, the limiting translation invariant behavior of WSTs (see Theorem 2.2.17) have not been extended beyond the wavelet case. Moreover, the result for WSTs is only known to hold if the corresponding wavelets are admissible (see Definition 2.2.7)

In this chapter, we study sequences of general scattering transforms for which the supports of the output generating functions are converging to $\{0\}$, and we prove a similar translation-invariant behavior in the limit as in the WST case. This allows for the removal of the requirement of the admissibility condition for the WST as long as the father wavelet is bandlimited. We end this chapter with a particular application of our results to sequences of Fourier scattering transforms. This chapter is based off work with Wojciech Czaja and David Koralov from [43].

3.2 Sequences of General Scattering Transforms

In [88], the near-translation invariance is interpreted in terms of the limiting behavior of $S[\mathcal{P}_{W,J}]$ as $J \to \infty$. Namely, Mallat considers a sequence of scattering transforms depending on a parameter J that defines the size of the central element ϕ_J. We introduce a corresponding notion for general constructions of scattering transforms.

Definition 3.2.1. *Let Λ_J be a sequence of countable indexing sets with $0 \neq \Lambda_J$ for all J. For each J, let $\mathcal{G}_J = \{g_{0,J}\} \cup \{g_\lambda\}_{\lambda \in \Lambda_J} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a sequence of semi-discrete Bessel sequences with upper frame bounds 1. Suppose further that each $g_{0,J}$ is bandlimited.*

We call the sequence of scattering transforms $S[\mathcal{P}_J]$ coherent, *where $\mathcal{P}_J = \bigcup_{k=0}^{\infty} \Lambda_J^k$ is the set of paths corresponding to the semi-discrete Bessel Sequences $\mathcal{G}_J$, $J \in \mathbb{N}$, if*

1. *$\mathcal{G}_J \setminus \{g_{0,J}\} \subseteq \mathcal{G}_{J+1} \setminus \{g_{0,J+1}\}$ for each J;*
2. *$\lim_{J\to\infty} \sup\{|\xi| : \xi \in \text{supp}(\widehat{g_{0,J}})\} = 0$.*

Remark 3.2.2. *The index sets Λ_J for different J are selected to be consistent with each other, in particular, $\Lambda_J \subseteq \Lambda_{J+1}$. The only element of $\mathcal{G}_J$ that might not be in $\mathcal{G}_{J+1}$ is the output-generating function $g_{0,J}$.*

We denote $D_J \coloneqq \sup\{|\xi| : \xi \in \text{supp}(\widehat{g_{0,J}})\}$, and we denote the scattering propagators by $U_J[p]$ for $p \in \mathcal{P}_J$. Note that the sequence of wavelet scattering transforms constructed using $\{\phi_J\} \cup \{\psi_\lambda : \lambda \in \Lambda_{W,J}\}$, $J = 1, 2, \ldots$ (see Section 2.2.3.1) is a coherent sequence, with $g_{0,J} = \phi_J$, as long as ϕ is bandlimited.

3.3 Translation Invariance in the Limit of General Scattering Transforms

Recall that Mallat's result, Theorem 2.2.17, states that, for wavelets satisfying the admissibility condition (Definition 2.2.7) and for each $f \in L^2(\mathbb{R}^d)$ and $c \in \mathbb{R}^d$,

$$\lim_{J\to\infty} \|S[P_{W,J}]f - S[P_{W,J}]T_c f\|_{l^2L^2} = 0.$$

In this section, we shall state and prove the corresponding result for the class of all coherent scattering transform sequences, without requiring an admissibility condition. Note that this result also proves Theorem 2.2.17 for wavelets when the father limit ϕ is band-limited. Thus, in some cases, we have demonstrated that Mallat's admissibility condition is not required for this result. First, we have the following main result.

Theorem 3.3.1. *Let $S[\mathcal{P}]$ be a scattering transform with an index set Λ and an output-generating function g_0. Let $D = \sup\{|\xi| : \xi \in \text{supp}(\widehat{g_{0,J}})\}$, and assume $D < \infty$. Then, for each $c \in \mathbb{R}^d$,*

$$\|S[\mathcal{P}]f - S[\mathcal{P}]T_c f\|_{l^2L^2} \leq 2\pi D|c| \, \|f\|_{L^2} .$$

Proof. Observe that

$$\|S[\mathcal{P}]f - S[\mathcal{P}]T_c f\|^2_{l^2L^2} = \sum_{p\in\mathcal{P}} \|U[p]f * g_0 - U[p](T_c f) * g_0\|^2_{L^2} .$$

Since the translation operator T_c commutes both with the modulus operator and with the convolution with g_λ for each $\lambda \in \Lambda$, we have $U[p](T_c f) = T_c U[p]f$. Without loss of generality, assume that $\|f\|_{L^2} \neq 0$.

By the Plancherel Theorem, for each $p \in \mathcal{P}$, we have

$$\|U[p]f * g_0 - (T_c U[p]f) * g_0\|_{L^2}^2 = \int_{\mathbb{R}^d} \left| \widehat{U[p]f}(\xi)\widehat{g_0}(\xi) - e^{-2\pi i \xi \cdot c} \widehat{U[p]f}(\xi)\widehat{g_0}(\xi) \right|^2 \mathrm{d}\xi.$$

Next, notice that for any real number r, $|e^{ir} - 1| \leq |r|$. Hence, for each $\xi \in B_D(0)$, since $|\xi \cdot c| \leq D|c|$,

$$|e^{-2\pi i \xi \cdot c} - 1| \leq 2\pi D|c|.$$

Thus, since $\mathrm{supp}(\widehat{g_0}) \subseteq \overline{B_D(0)}$, we have

$$\begin{aligned}\int_{\mathbb{R}^d} \left| \widehat{U[p]f}(\xi)\widehat{g_0}(\xi) - e^{-2\pi i \xi \cdot c} \widehat{U[p]f}(\xi)\widehat{g_0}(\xi) \right|^2 \mathrm{d}\xi &= \int_{\overline{B_D(0)}} \left| \widehat{U[p]f}(\xi)\widehat{g_0}(\xi) \right|^2 \left| 1 - e^{-2\pi i \xi \cdot c} \right|^2 \mathrm{d}\xi \\ &\leq (2\pi D|c|)^2 \left\| \widehat{U[p]f}\widehat{g_0} \right\|_{L^2}^2.\end{aligned}$$

Moreover, since scattering transforms are non-expansive (Corollary 2.2.14), we have

$$\|S[\mathcal{P}]f\|_{l^2 L^2}^2 = \sum_{p \in \mathcal{P}} \|U[p]f * g_0\|_{L^2}^2 = \sum_{p \in P} \left\| \widehat{U[p]f}\widehat{g_0} \right\|_{L^2}^2 \leq \|f\|_{L^2}^2 .$$

Hence, we have

$$\begin{aligned}\|S[\mathcal{P}]f - S[\mathcal{P}]T_c f\|_{l^2 L^2}^2 &= \sum_{p \in \mathcal{P}} \|U[p]f * g_0 - (T_c U[p]f) * g_0\|_{L^2}^2 \\ &\leq (2\pi D|c|)^2 \sum_{p \in \mathcal{P}} \left\| \widehat{U[p]f}\widehat{g_0} \right\|_{L^2}^2 \leq (2\pi D|c|)^2 \|f\|_{L^2}^2 .\end{aligned}$$

□

As a corollary, we get limiting translation invariance for coherent sequences of scattering transforms.

Corollary 3.3.2. *For any coherent sequence of scattering transforms $S[\mathcal{P}_J]$, for each $f \in L^2(\mathbb{R}^d)$, and each $c \in \mathbb{R}^d$,*

$$\lim_{J\to\infty} \|S[\mathcal{P}_J]f - S[\mathcal{P}_J]T_c f\|_{l^2L^2} = 0.$$

Proof. This is obtained directly by applying the bound from Theorem 3.3.1, which states that for each J,

$$\|S[\mathcal{P}_J]f - S[\mathcal{P}_J]T_c f\|^2_{l^2L^2} \leq (2\pi D_J|c|)^2 \|f\|^2_{L^2}.$$

Note that $D_J \to 0$, while all the other factors on the right-hand side do not depend on J. Hence, as $J \to \infty$, the right hand side goes to 0. □

3.4 Applications to Sequences of Fourier Scattering Transforms

Our original motivation for writing the paper was to prove Corollary 3.3.2 for coherent sequences of Fourier scattering transforms [44] (see Section 2.2.3.2 for the definition of such transforms). The following result was previously known for norm differences under translation for the Fourier scattering transform.

Theorem 3.4.1. (Theorem 3.6 Part (d) from [44]) *There exists $C > 0$, depending only on the uniform covering frame $\mathcal{G}_F = \{g_0\} \cup \{g_\lambda\}_{\lambda\in\Lambda_F}$, such that for all $f \in L^2(\mathbb{R}^d)$ and $c \in \mathbb{R}^d$,*

$$\|S[\mathcal{P}_F]f - S[\mathcal{P}_F]T_c f\|_{l^2L^2} \leq C|c| \|\nabla g_0\|_{L^1} \|f\|_{L^2}.$$

Applying the bound in Theorem 3.3.1, we can provide an alternative to the upper bound in Theorem 3.4.1:

$$\|S[\mathcal{P}_F]f - S[\mathcal{P}_F]T_c f\|_{l^2L^2} \leq 2\pi D_F|c| \, \|f\|_{L^2},$$

where $D_F = \sup\{|\xi| : \xi \in \text{supp}(\widehat{g_0})\}$. Since $\mathcal{G}_F$ is a uniform covering frame, we know that $D_F < \infty$. So, applying Corollary 3.3.2 to the Fourier scattering transform, we obtain the following result.

Corollary 3.4.2. *Suppose $\{\mathcal{G}_{F,J}\}_{J\in\mathbb{N}}$ is a coherent sequence of Uniform Covering Frames, with index sets $\Lambda_{F,J}$, output-generating functions $g_{0,J}$, and with the set of paths $P_{F,J} \coloneqq \bigcup_{k=0}^{\infty} \Lambda_J^k$. Let $S[\mathcal{P}_{F,J}]$ be the corresponding Fourier scattering transforms. Then, for every $f \in L^2(\mathbb{R}^d)$, and for every $c \in \mathbb{R}^d$,*

$$\lim_{J\to\infty} \|S[P_{F,J}]f - S[P_{F,J}]T_c f\|_{l^2L^2} = 0.$$

Thus, we have been able to show an analogous result to Mallat's translation invariance in the limit of WSTs for FSTs.

Chapter 4: Stationary Scattering Transforms

4.1 Introduction

While scattering transforms have been well-studied and heavily applied in problems with deterministic inputs, some inquiry into how these objects act on random data has also been conducted. In his original paper [88], Mallat extended the construction of wavelet scattering transforms to also act on strictly stationary processes and proved several properties analogous to the deterministic case. Additionally, he and his students considered applications of the WST to the analysis and classification of textures, which can be modeled as instances of strictly stationary processes [24,26,86,114,115]. However, there has not been research on how scattering transforms with non-wavelet based filters behave when acting on such processes.

In this chapter, we extend the definition of scattering transforms on strictly stationary processes to use more general families of convolutional filters. We also prove that stationary scattering transforms have similar properties as in the deterministic case, such as non-expansiveness, limiting translation invariance behavior, and stability to small (random) deformations. We also consider a specific case in which the convolutional filters are uniform covering frames, and prove that the stationary Fourier scattering transform also has layer energy which decays exponentially fast. This chapter is based on joint work I conducted with Wojciech Czaja and David Koralov from [42].

4.2 Definitions

First, we shall define the stationary scattering transform. Let Λ be an a countable indexing set such that $0 \notin \Lambda$, and let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a semi-discrete Bessel sequence with upper frame bound $B \leq 1$ (see Definition 2.2.1). That is, by Equation 2.6, we have for all $\xi \in \mathbb{R}^d$ that

$$|\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda} |\widehat{g_\lambda}(\xi)|^2 \leq 1.$$

For $k \in \mathbb{N}$, let $\Lambda^k := \overbrace{\Lambda \times \cdots \times \Lambda}^{k \text{ times}}$, and define $\Lambda^0 := \{\varnothing\}$. Let $\mathcal{P} = \bigcup_{k=0}^{\infty} \Lambda^k$ be the set of all paths. We next define the stationary scattering propagator (cf. Definition 2.2.2).

Definition 4.2.1. *Let $p \in \Lambda^k$ be a path and let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a mean square continuous, strictly stationary process. The stationary scattering propagator $U[p]$ of X is defined by*

$$U[p]X(\omega, t) = \begin{cases} X(\omega, t), & k = 0, \\ \left|U[(\lambda_1, \ldots, \lambda_{k-1})]X * g_{\lambda_k}(\omega, t)\right|, & k \geq 1. \end{cases}$$

We now define the stationary scattering transform (cf. Definition 2.2.3).

Definition 4.2.2. *For $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$, a mean square continuous, strictly stationary process, the stationary scattering transform $S[\mathcal{P}]$ of X is defined by*

$$S[\mathcal{P}]X(\omega, t) = \{U[p]X * g_0(\omega, t) : p \in \mathcal{P}\}. \tag{4.1}$$

Note that, since convolutions and modulations preserve strict stationarity and mean square con-

tinuity, $U[p]X$ and $U[p]X * g_0$ are strictly stationary and mean square continuous for all paths $p \in \mathcal{P}$. We define the *energy* of $S[\mathcal{P}]X$ as

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right] \coloneqq \sum_{p\in\mathcal{P}} \mathbb{E}\left[|U[p]X * g_0(\cdot,t)|^2\right]. \tag{4.2}$$

This is precisely the square of the $l^2L^2(\mathcal{P};\Omega)$ norm of $S[\mathcal{P}]X(\cdot,t)$. Note that, by strict stationarity, all expectations are independent of $t \in \mathbb{R}^d$. By Corollary 4.3.2, we will see that $S[\mathcal{P}]$ maps a mean square continuous, strictly stationary process X to a sequence of mean square continuous, strictly stationary processes satisfying $\mathbb{E}\left[S[\mathcal{P}]X(\cdot,t)|^2\right] < \infty$.

Remark 4.2.3. *Many of the further generalizations Wiatowski and Bölcskei proposed for the deterministic case can also be used in this setting. Indeed, one could easily extend Definitions 4.2.1 and 4.2.2 to use different semi-discrete Bessel sequences for each layer. However, the authors also replaced the modulus with general, nonlinear Lipchitz operators $M_k : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ (where k is the layer number) and also considered applying pooling operators $P_k : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ after applying M_k. Since the only requirements on these operators is that they are Lipschitz, it is not in general true that these operators would preserve strict stationarity.*

At the same time, Wiatowski and Bölcskei also considered specific examples of M_k and P_k which could be used to extend our definitions. For M_k, they considered operators of the form $M_k f = \sigma_k(f)$, where $\sigma_k : \mathbb{C} \to \mathbb{C}$ is a pointwise, Lipschitz continuous function. As such functions are measurable, they would preserve strict stationarity stationarity and could thus be used in place of the modulus. For P_k, they specifically considered convolution operators, which also preserve strict stationarity by Proposition 2.3.4.

4.3 Properties of the Stationary Scattering Transform

In this section, we prove several properties of the stationary scattering transform, analogous to the properties of the deterministic scattering transform. Namely, we prove that the stationary scattering transform is non-expansive (i.e., Lipschitz), mean-square continuous with respect to translations in t, and stable with respect to small, random deformations.

4.3.1 Non-Expansiveness

First, we prove that the stationary scattering transform is non-expansive (cf. Theorem 2.2.12).

Theorem 4.3.1. *For any jointly strictly stationary processes $X, Y : \Omega \times \mathbb{R}^d \to \mathbb{C}$ which are mean square continuous, we have*

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]Y(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X(\cdot,t) - Y(\cdot,t)|^2\right] \text{ for all } t \in \mathbb{R}^d. \tag{4.3}$$

Proof. Fix a path $p \in \Lambda^k$, and define $X_p(\omega,t) \coloneqq U[p]X(\omega,t)$ and $Y_p(\omega,t) \coloneqq U[p]Y(\omega,t)$. Following a similar proof as in Proposition 2.3.4 part (2), we see that X_p and Y_p are jointly strictly stationary for all p.

For $s \in \mathbb{R}^d$, let $R_p(s) = \mathbb{E}\left[(X_p - Y_p)(\cdot, s+t)\overline{(X_p - Y_p)(\cdot,t)}\right]$ denote the autocorrelation function of $X_p - Y_p$. By Bochner's Theorem (Theorem 2.3.2), since $X_p - Y_p$ is strictly stationary and mean

square continuous, there exists a bounded Borel measure μ_p such that

$$R_p(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} \mathrm{d}\mu_p(\xi).$$

In particular, we have

$$\mathbb{E}\left[|X_p(\cdot,t) - Y_p(\cdot,t)|^2\right] = R_p(0) = \int_{\mathbb{R}^d} \mathrm{d}\mu_p(\xi).$$

Next, fix $\lambda_{k+1} \in \Lambda$ and note that

$$\begin{aligned} |U[\lambda_{k+1}]X_p(\omega,t) - U[\lambda_{k+1}]Y_p(\omega,t)| &= \left||X_p * g_{\lambda_{k+1}}(\omega,t)| - |Y_p * g_{\lambda_{k+1}}(\omega,t)|\right| \\ &\leq |X_p * g_{\lambda_{k+1}}(\omega,t) - Y_p * g_{\lambda_{k+1}}(\omega,t)|, \end{aligned}$$

P-a.s. for each $t \in \mathbb{R}^d$ by the triangle inequality. Using this inequality and Proposition 2.3.3 part (5), we have that

$$\begin{aligned} &\mathbb{E}\left[|X_p * g_0(\cdot,t) - Y_p * g_0(\cdot,t)|^2\right] + \sum_{\lambda_{k+1}\in\Lambda} \mathbb{E}\left[|U[\lambda_{k+1}]X_p(\cdot,t) - U[\lambda_{k+1}]Y_p(\cdot,t)|^2\right] \\ &\leq \mathbb{E}\left[|X_p * g_0(\cdot,t) - Y_p * g_0(\cdot,t)|^2\right] + \sum_{\lambda_{k+1}\in\Lambda} \mathbb{E}\left[|X_p * g_{\lambda_{k+1}}(\cdot,t) - Y_p * g_{\lambda_{k+1}}(\cdot,t)|^2\right] \\ &= \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 \mathrm{d}\mu_p(\xi) + \sum_{\lambda_{k+1}\in\Lambda} \int_{\mathbb{R}^d} |\widehat{g_{\lambda_{k+1}}}(\xi)|^2 \mathrm{d}\mu_p(\xi) \\ &\leq \int_{\mathbb{R}^d} 1 d\mu_p(\xi) = \mathbb{E}\left[|X_p(\cdot,t) - Y_p(\cdot,t)|^2\right]. \end{aligned}$$

Summing over all $p \in \Lambda^k$ and recalling that every $q \in \Lambda^{k+1}$ can be decomposed as $q = (p, \lambda_{k+1})$ for

$p \in \Lambda^k$ and $\lambda_{k+1} \in \Lambda$, we can rewrite the above as

$$\begin{aligned}
&\sum_{p \in \Lambda^k} \mathbb{E}\left[|U[p]X * g_0(\cdot, t) - U[p]Y * g_0(\cdot, t)|^2\right] + \sum_{q \in \Lambda^{k+1}} \mathbb{E}\left[|U[q]X(\cdot, t) - U[q]Y(\cdot, t)|^2\right] \\
&\leq \sum_{p \in \Lambda^k} \mathbb{E}\left[|U[p]X(\cdot, t) - U[p]Y(\cdot, t)|^2\right].
\end{aligned}$$

Rearranging and summing over paths of lengths k from 0 to K, we get by a telescoping argument that

$$\begin{aligned}
&\sum_{k=0}^{K} \sum_{p \in \Lambda^k} \mathbb{E}\left[|U[p]X * g_0(\cdot, t) - U[p]Y * g_0(\cdot, t)|^2\right] \\
&\leq \sum_{k=0}^{K} \left(\sum_{p \in \Lambda^k} \mathbb{E}\left[|U[p]X(\cdot, t) - U[p]Y(\cdot, t)|^2\right] - \sum_{p \in \Lambda^{k+1}} \mathbb{E}\left[|U[p]X(\cdot, t) - U[p]Y(\cdot, t)|^2\right] \right) \\
&= \sum_{p \in \Lambda^0} \mathbb{E}\left[|U[p]X(\cdot, t) - U[p]Y(\cdot, t)|^2\right] - \sum_{p \in \Lambda^{K+1}} \mathbb{E}\left[|U[p]X(\cdot, t) - U[p]Y(\cdot, t)|^2\right] \\
&\leq \mathbb{E}\left[|X(\cdot, t) - Y(\cdot, t)|^2\right],
\end{aligned}$$

given that $\Lambda^0 = \{\varnothing\}$. Thus, we see that

$$\begin{aligned}
\mathbb{E}(|S[\mathcal{P}]X(\cdot, t) - S[\mathcal{P}]Y(\cdot, t)|^2) &= \lim_{K \to \infty} \sum_{k=0}^{K} \sum_{p \in \Lambda^k} \mathbb{E}\left[|U[p]X * g_0(\cdot, t) - U[p]Y * g_0(\cdot, t)|^2\right] \\
&\leq \mathbb{E}\left[|X(\cdot, t) - Y(\cdot, t)|^2\right]
\end{aligned}$$

□

As an immediate consequence, the stationary scattering transform is bounded (cf. Corollary 2.2.14).

Corollary 4.3.2. *For all strictly stationary processes $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ which are mean square*

continuous, we have

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X(\cdot,t)|^2\right], \text{ for all } t \in \mathbb{R}^d. \tag{4.4}$$

Proof. Apply Theorem 4.3.1 with $Y \equiv 0$. □

4.3.2 Mean Square Continuity

As mentioned previously, every individual output $U[p]X * g_0$ of the stationary scattering transform is mean square continuous. However, we can prove a stronger statement that the stationary scattering transform is mean square continuous with respect to the energy defined by Equation 4.2. Moreover, we can prove results on how fast this energy goes to 0 as the shift s goes to 0.

Theorem 4.3.3. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a mean square continuous, strictly stationary process.*

(1) For all $t \in \mathbb{R}^d$,

$$\lim_{s \to 0} \mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] = 0. \tag{4.5}$$

That is, the strictly stationary process $S[\mathcal{P}]X$ (which takes values in the Hilbert space $l^2(\mathcal{P})$) is mean square continuous.

(2) Suppose $g_0 \neq 0$. There exists a Borel measure $\widetilde{\mu}$ (depending on X) such that, for all $R > 0$, if $s \in \mathbb{R}^d$ with $|s_1| < \frac{1}{4R}$ and $s_i = 0$ for $i \neq 1$, then

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] \geq \pi^2 |s_1|^2 \int\limits_{B_R(0)} |\xi_1|^2 |\widehat{g_0}(\xi)|^2 d\widetilde{\mu}(\xi). \tag{4.6}$$

In particular, if $\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right]$ is not identically 0 in s, then Equation 4.5

can't converge to 0 *faster than quadratically in* $|s|$ *as* $s \to 0$.

(3) Suppose g_0 *is bandlimited (i.e.,* $\widehat{g_0}$ *has compact support), and let* $D \coloneqq \sup\{|\xi| : \xi \in supp(\widehat{g_0})\}$. *Then, for all* $t, s \in \mathbb{R}^d$,

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] \leq 4\pi^2 D^2 |s|^2 \mathbb{E}\left[|X(\cdot,t)|^2\right]. \tag{4.7}$$

Thus, the limit in Equation 4.5 decays to 0 *quadratically in* $|s|$.

Proof. Proof of Part (1): Similarly to the proof of Theorem 4.3.1, for each $p \in \mathcal{P}$, let $X_p(\omega,t) = U[p]X(\omega,t)$ and $R_p(s) = \mathbb{E}\left[X_p(\cdot,s+t)\overline{X_p(\cdot,t)}\right]$ be the autocorrelation function of X_p. By Theorem 2.3.2, there exists a bounded Borel measure μ_p such that

$$R_p(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} \mathrm{d}\mu_p(\xi) \text{ for all } s \in \mathbb{R}^d.$$

Similarly, define $R_{p,g_0}(s) = \mathbb{E}\left[X_p * g_0(\cdot,s+t)\overline{X_p * g_0(\cdot,t)}\right]$. Then, by Proposition 2.3.3 part (5), we have that

$$R_{p,g_0}(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} |\widehat{g_0}(\xi)|^2 \mathrm{d}\mu_p(\xi) \text{ for all } s \in \mathbb{R}^d.$$

By Corollary 4.3.2, we know that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X(\cdot,t)|^2\right] < \infty, \text{ for all } t \in \mathbb{R}^d.$$

By definition, we have that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right] = \sum_{p\in\mathcal{P}} \mathbb{E}\left[|U[p]X * g_0(\cdot,t)|^2\right] = \sum_{p\in\mathcal{P}} R_{p,g_0}(0) = \sum_{p\in\mathcal{P}} \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 \mathrm{d}\mu_p(\xi)$$

Define the set function $\widetilde{\mu}$ on $\mathcal{B}(\mathbb{R}^d)$ by $\widetilde{\mu}(A_d) = \sum_{p\in\mathcal{P}} \mu_p(A_d)$ for $A_d \in \mathcal{B}(\mathbb{R}^d)$. Since the μ_p are all non-negative measures, it is easy to see that $\widetilde{\mu}$ is a measure and hence that $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d), \widetilde{\mu})$ is a measure space. Moreover, since $\sum_{p\in\mathcal{P}} \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 \mathrm{d}\mu_p(\xi) < \infty$ and the terms are non-negative, we see that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right] = \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 \mathrm{d}\widetilde{\mu}(\xi) < \infty. \tag{4.8}$$

Next, for fixed $p \in \mathcal{P}$ note that

$$\begin{aligned}
&\mathbb{E}\left[|X_p * g_0(\cdot,t) - X_p * g_0(\cdot,t+s)|^2\right] \\
&= \mathbb{E}\left[|X_p * g_0(\cdot,t)|^2\right] - 2\mathrm{Re}\left(\mathbb{E}\left[X_p * g_0(\cdot,t+s)\overline{X_p * g_0(\cdot,t)}\right]\right) + \mathbb{E}\left[|X_p * g_0(\cdot,t+s)|^2\right] \\
&= 2R_{p,g_0}(0) - 2\mathrm{Re}\left(R_{p,g_0}(s)\right) = \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 (2 - 2\cos(2\pi\xi\cdot s)) \mathrm{d}\mu_p(\xi)
\end{aligned}$$

by Euler's formula. Summing over all $p \in \mathcal{P}$, we thus have that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] = \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 (2 - 2\cos(2\pi\xi\cdot s)) \mathrm{d}\widetilde{\mu}(\xi),$$

which is well-defined because $2 - 2\cos(2\pi\xi\cdot s)$ is bounded and $|\widehat{g_0}|^2$ is integrable with respect to $\widetilde{\mu}$. Clearly, $\lim_{s\to 0} 2 - 2\cos(2\pi\xi\cdot s) = 0$ for all $\xi \in \mathbb{R}^d$. Thus, by the Lebesgue dominated convergence t

heorem, we have that

$$\lim_{s\to 0}\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)-S[\mathcal{P}]X(\cdot,t+s)|^2\right]=0$$

for all $t\in\mathbb{R}^d$. This proves part (1).

Proof of Part (2): Let $R>0$ be arbitrary, and let $s\in\mathbb{R}^d$ such that $s_2=\cdots=s_d=0$. Note that $2-2\cos(2\pi\xi\cdot s)=4\sin^2(\pi\xi\cdot s)$ by a standard trigonometry identity. It is well known that, for all $-\frac{\pi}{2}\le x\le\frac{\pi}{2}$, we have $|\sin(x)|\ge\frac{|x|}{2}$. Thus, if $|\pi\xi\cdot c|\le\frac{\pi}{2}$, we have that $4\sin^2(\pi\xi\cdot s)\ge|\pi\xi\cdot s|^2=\pi^2|\xi_1 s_1|^2$.

Let $\delta=\frac{1}{4R}$ so that, if $|s|=|s_1|<\delta$ and $\xi\in B_R(0)$, then $|\xi\cdot s|\le|\xi||s|<\frac{1}{4}<\frac{1}{2}$. Hence, we see that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)-S[\mathcal{P}]X(\cdot,t+s)|^2\right]=\int_{\mathbb{R}^d}|\widehat{g_0}(\xi)|^2 4\sin^2(\pi\xi\cdot s)\mathrm{d}\widetilde{\mu}(\xi)$$

$$\ge\int_{B_R(0)}|\widehat{g_0}(\xi)|^2 4\sin^2(\pi\xi\cdot s)\mathrm{d}\widetilde{\mu}(\xi)\ge\pi^2|s_1|^2\int_{B_R(0)}|\widehat{g_0}(\xi)|^2|\xi_1|^2\mathrm{d}\widetilde{\mu}(\xi).$$

Now, suppose that $\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)-S[\mathcal{P}]X(\cdot,t+s)|^2\right]$ is not identically 0 in s. I claim that there must exist some $0<R_1<R_2<\infty$ such that

$$\int_{B_{R_2}(0)\setminus B_{R_1}(0)}|\widehat{g_0}(\xi)|^2\mathrm{d}\widetilde{\mu}(\xi)>0. \tag{4.9}$$

Suppose not. Then, by sending $R_2\to\infty$ and $R_1\to 0$, we will have by the Lebesgue dominated convergence theorem that

$$\int_{\mathbb{R}^d\setminus\{0\}}|\widehat{g_0}(\xi)|^2\mathrm{d}\widetilde{\mu}(\xi)=0.$$

Since $|\widehat{g_0}| \geq 0$, we see that the measure $|\widehat{g_0}|^2 \mathrm{d}\widetilde{\mu}$ is only supported on the set $\{0\}$. This clearly implies that

$$\int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} |\widehat{g_0}(\xi)|^2 \mathrm{d}\widetilde{\mu}(\xi) = e^{2\pi i (0) \cdot s} |\widehat{g_0}(0)|^2 \widetilde{\mu}(\{0\}) = |\widehat{g_0}(0)|^2 \widetilde{\mu}(\{0\}), \tag{4.10}$$

for all $s \in \mathbb{R}^d$.

Now, define the autocorrelation function $R_{S[\mathcal{P}]X}(s)$ of $S[\mathcal{P}]X$ by

$$R_{S[\mathcal{P}]X}(s) \coloneqq \sum_{p \in \mathcal{P}} R_{p,g_0}(s) = \int_{\mathbb{R}^d} e^{2\pi i \xi \cdot s} |\widehat{g_0}(\xi)|^2 \mathrm{d}\widetilde{\mu}(\xi) = |\widehat{g_0}(0)|^2 \widetilde{\mu}(\{0\}),$$

for all $s \in \mathbb{R}^d$. Next, by expanding the terms, we see that

$$\begin{aligned}
&\mathbb{E}\left[|S[\mathcal{P}]X(\cdot, t) - S[\mathcal{P}]X(\cdot, t+s)|^2\right] = \sum_{p \in \mathcal{P}} \mathbb{E}\left[|X_p * g_0(\cdot, t) - X_p * g_0(\cdot, t+s)|^2\right] \\
=&\sum_{p \in \mathcal{P}} 2R_{p,g_0}(0) - 2\mathrm{Re}(R_{p,g_0}(s)) = 2R_{S[\mathcal{P}]X}(0) - 2\mathrm{Re}(R_{S[\mathcal{P}]X}(s)) \\
=&2|\widehat{g_0}(0)|^2 \widetilde{\mu}(\{0\}) - 2|\widehat{g_0}(0)|^2 \widetilde{\mu}(\{0\}) = 0,
\end{aligned}$$

for all $s \in \mathbb{R}^d$, contradicting that $\mathbb{E}\left[|S[\mathcal{P}]X(\cdot, t) - S[\mathcal{P}]X(\cdot, t+s)|^2\right]$ is not identically 0.

So, there exists $0 < R_1 < R_2 < \infty$ such that Equation 4.9 holds. Taking $R = R_2$, we must have that

$$\int_{B_R(0)} |\xi_1|^2 |\widehat{g_0}(\xi)|^2 \mathrm{d}\widetilde{\mu}(\xi) > 0.$$

Thus, combining this with Equation 4.6 tells us that there exists a constant $C > 0$ such that, for all

$s \in \mathbb{R}^d$ with $|s_1| < \frac{1}{4\widetilde{R}}$ and $s_i = 0$ for $i \neq 1$, we have

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] \geq C|s_1|^2.$$

Thus, the left hand side can't converge to 0 faster than quadratically in $|s|$.

Proof of Part (3): Now, suppose that g_0 is bandlimited. As in the proof of part (1), we can write

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] = \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 4\sin^2(\pi\xi\cdot s)\mathrm{d}\widetilde{\mu}(\xi)$$

For all $x \in \mathbb{R}$, it is well known that $|\sin(x)| \leq |x|$. Hence, for all $\xi \in \text{supp}(\widehat{g_0})$, we have that $\sin^2(\pi\xi\cdot s) \leq \pi^2|\xi\cdot s|^2 \leq \pi^2 D^2|s|^2$. Thus, we see that

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X(\cdot,t+s)|^2\right] \leq 4\pi^2D^2|s|^2\int_{\mathbb{R}^d}|\widehat{g_0}(\xi)|^2\mathrm{d}\widetilde{\mu}(\xi) = 4\pi^2D^2|s|^2\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t)|^2\right]$$

Applying Corollary 4.3.2, we get Inequality 4.7. □

Remark 4.3.4. *Note that Inequality 4.7 of this theorem is a similar result as in Theorem 3.3.1. However, the two results are used for different purposes; for the latter, we used it show that the limit of the sequences of scattering transforms was translation invariant, while the for the former we only showed that the scattering transform was mean square continuous as the translations went to* 0.

4.3.3 Stability under Small Deformations

In this section, we will prove that stationary scattering transforms are stable under the actions of small (independent, random) deformations.

4.3.3.1 Preserving Strict Stationarity

First, we will prove results on when these actions preserve strict stationarity. :et $\tau : \Omega \times \mathbb{R}^d \to \mathbb{R}^d$ be strictly stationary such that, P-a.s., $\tau(\omega, \cdot)$ is a $C^1(\mathbb{R}^d; \mathbb{R}^d)$ function in $t \in \mathbb{R}^d$ satisfying $||\nabla\tau(\omega, \cdot)||_{L^\infty(\mathbb{R}^d)} < 1$ (where $\nabla\tau$ denotes the differential of τ in the second variable). In particular, P-a.s. $t \mapsto t - \tau(\omega, t)$ is a diffeomorphism. On page 1371 of [88], it was claimed that if X, τ are independent, strictly stationary process with τ satisfying the assumptions above then $X_\tau(\omega, t) \coloneqq X(\omega, t - \tau(\omega, t))$ is itself a strictly stationary process. This is relatively easy to show if τ P-a.s. takes at most countably many values, i.e., there exists a countable set $\mathcal{A} \subseteq \mathbb{R}^d$ such that $P(\tau(\cdot, t) \in \mathcal{A}) = 1$ for all $t \in \mathbb{R}^d$; see Proposition 4.3.5. Note that, in this case, the assumption that τ P-a.s. has continuously differentiable realizations will imply that τ is P-a.s. constant in t.

Proposition 4.3.5. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ and $\tau : \Omega \times \mathbb{R}^d \to \mathbb{R}^d$ be strictly stationary processes such that X is mean square continuous and P-a.s. $\tau(\omega, \cdot)$ is a $C^1(\mathbb{R}^d; \mathbb{R}^d)$ function in t satisfying $||\nabla\tau(\omega, \cdot)||_{L^\infty(\mathbb{R}^d)} < 1$. Suppose further that X and τ are independent and that τ P-a.s. takes countably many values. Then,*

(1) $X_\tau(\omega, t) \coloneqq X(\omega, t - \tau(\omega, t))$ *is a strictly stationary process, for $\omega \in \Omega, t \in \mathbb{R}^d$;*

(2) $\mathbb{E}[|X_\tau(\cdot, t)|] = \mathbb{E}[X(\cdot, t)]$ *for $t \in \mathbb{R}^d$;*

(3) $\mathbb{E}[|X_\tau(\cdot, t)|^2] = \mathbb{E}[|X(\cdot, t)|^2]$ *for $t \in \mathbb{R}^d$;*

(4) $\lim_{s \to 0} \mathbb{E}[|X_\tau(\cdot, t) - X_\tau(\cdot, t + s)|^2] = 0$ *for all $t \in \mathbb{R}^d$;*

(5) *X and X_τ are jointly strictly stationary.*

Proof. We assume that $\tau(t)$ has countably many values, i.e., there exists a countable set $\mathcal{A} = \{a_m : m \in \mathbb{N}\} \subseteq \mathbb{R}^d$ such that $P(\tau(\cdot, t) \in \mathcal{A}) = 1$ for all $t \in \mathbb{R}^d$.

Fix $t_1, \ldots, t_n \in \mathbb{R}^d$ and $A_1, \ldots, A_n \in \mathcal{B}(\mathbb{C})$. Then, we have that:

$$\begin{aligned} &P\left(X_\tau(\cdot, t_i) \in A_i, i = 1, \ldots, n\right) \\ &= \sum_{m_1, \ldots, m_n = 1}^{\infty} P\big(X(\cdot, t_i - a_{m_i}) \in A_i \text{ and } \tau(\cdot, t_i) = a_{m_i}, i = 1, \ldots, n\big). \end{aligned}$$

Using independence and strict stationarity of X and τ, we see that this is equal to

$$\begin{aligned} &\sum_{m_1, \ldots, m_n = 1}^{\infty} P\big(X(\cdot, t_i - a_{m_i}) \in A_i, i = 1, \ldots, n\big) P\big(\tau(\cdot, t_i) = a_{m_i}, i = 1, \ldots, n\big) \\ =&\sum_{m_1, \ldots, m_n = 1}^{\infty} P\big(X(\cdot, t_i + t - a_{m_i}) \in A_i, i = 1, \ldots, n\big) P\big(\tau(\cdot, t_i + t) = a_{m_i}, i = 1, \ldots, n\big) \\ =&\sum_{m_1, \ldots, m_n = 1}^{\infty} P\big(X(\cdot, t_i + t - a_{m_i}) \in A_i, \tau(\cdot, t_i + t) = a_{m_i}, i = 1, \ldots, n\big) \\ =&P(X_\tau(\cdot, t_i + t) \in A_i, i = 1 \ldots, n). \end{aligned}$$

Hence, X_τ is strictly stationary, proving part (1). The proof of part (5) follows by a similar argument.

For part (2), we perform a similar calculation:

$$\begin{aligned} \mathbb{E}\left[|X_\tau(\cdot, t)|\right] &= \sum_{m=1}^{\infty} \mathbb{E}\left[|X(\cdot, t - a_m) \mathbb{1}_{\tau(\cdot, t) = a_m}|\right] \\ &= \sum_{m=1}^{\infty} \mathbb{E}\left[|X(\cdot, t - a_m)|\right] P(\tau(\cdot, t) = a_m) \\ &= \mathbb{E}\left[|X(\cdot, t)|\right], \end{aligned}$$

by strict stationarity of X. Using the fact that X is mean square continuous, one can prove parts (3)

and (4) by a similar argument. □

However, X_τ need not be strictly stationary if τ takes more than countably many values, as shown by the following example.

Example 4.3.6. *Let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $Y : \Omega \times \mathbb{R} \to \mathbb{C}$ be a strictly stationary process such that $|Y(\omega, t)| \leq 1$ for all $\omega \in \Omega, t \in \mathbb{R}$. Let θ be uniformly distributed on $(0, 2\pi)$ and independent from Y, and define $\tau : \Omega \times \mathbb{R} \to \mathbb{R}$ by $\tau(\omega, t) := \frac{1}{2}\cos(t + \theta(\omega))$. It is easy to check that τ is strictly stationary and satisfies $\|\frac{d}{dt}\tau(\omega, \cdot)\|_{L^\infty(\mathbb{R}^d)} = \frac{1}{2}$ P-a.s.*

For $\omega \in A := \{\omega \in \Omega : \theta(\omega) \in [0, \frac{\pi}{2}]\}$, notice that $\tau(\omega, 0) \in [0, \frac{1}{2}]$. Define a new strictly stationary process $X : \Omega \times \mathbb{R} \to \mathbb{C}$ by

$$X(\omega, t) := \begin{cases} 2, & \omega \in A \text{ and } t = -\tau(\omega, 0), \\ Y(\omega, t), & \text{otherwise.} \end{cases}$$

For each $t \in \mathbb{R}$, notice that $P(\omega \in A, -\tau(\omega, 0) = t) = 0$ because θ is uniformly distributed. Hence, we see that $X(\omega, t) = Y(\omega, t)$ P-a.s. for each $t \in \mathbb{R}$. In particular, this implies that X remains strictly stationary and that X, τ are independent.

Now, consider X_τ; at $t = 0$, we have $X_\tau(\omega, 0) = X(\omega, -\tau(\omega, 0))$. If $\omega \in A$, then $X(\omega, -\tau(\omega, 0)) = 2$. However, for t sufficiently large, $t - \tau(\omega, t) > 2$ for all $\omega \in \Omega$ and so $|X(\omega, t - \tau(\omega, t))| = |Y(\omega, t - \tau(\omega, t))| \leq 1$. Since A has positive probability, this contradicts strict stationarity of X_τ, as $\mathbb{P}(X_\tau(\cdot, 0) = 2) > 0$ and $\mathbb{P}(X_\tau(\cdot, t) = 2) = 0$ for all all sufficiently large $t \in \mathbb{R}$.

In order to overcome the limitations posed by this counterexample, it will be enough to assume that P-a.s. the sample paths of X, i.e., the functions $t \mapsto X(\omega, t)$, are continuous; see Proposition 4.3.7

and cf. Proposition 4.3.5. Notice that, in the preceeding example, τ takes uncountably many values and X does not P-a.s. have continuous sample paths. Whether these are the only sets of conditions which ensure the results of these propositions, and in particular that X_τ is strictly stationary, remains an open problem.

Proposition 4.3.7. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ and $\tau : \Omega \times \mathbb{R}^d \to \mathbb{R}^d$ be strictly stationary processes such that X is mean square continuous and P-a.s. $\tau(\omega, \cdot)$ is a $C^1(\mathbb{R}^d; \mathbb{R}^d)$ function in t satisfying $||\nabla\tau(\omega, \cdot)||_{L^\infty(\mathbb{R}^d)} < 1$. Suppose further that X and τ are independent and that X P-a.s. has continuous sample paths. Then,*

(1) $X_\tau(\omega, t) \coloneqq X(\omega, t - \tau(\omega, t))$ *is a strictly stationary process, for $\omega \in \Omega, t \in \mathbb{R}^d$;*

(2) $\mathbb{E}[|X_\tau(\cdot, t)|] = \mathbb{E}[X(\cdot, t)]$ *for $t \in \mathbb{R}^d$;*

(3) $\mathbb{E}[|X_\tau(\cdot, t)|^2] = \mathbb{E}[|X(\cdot, t)|^2]$ *for $t \in \mathbb{R}^d$;*

(4) $\lim_{s \to 0} \mathbb{E}[|X_\tau(\cdot, t) - X_\tau(\cdot, t + s)|^2] = 0$ *for all $t \in \mathbb{R}^d$;*

(5) *X and X_τ are jointly strictly stationary.*

Proof. Assume that P-a.s. the realizations of X are continuous. Let $\mathcal{D}_{n,1} = \frac{1}{2^n}\mathbb{Z}^d$ be the n-th dyadic integer lattice, and define

$$\tau_n(\omega, t) \coloneqq \sum_{k \in \mathcal{D}_{n,1}} k \mathbb{1}_{B_{k,1}}(\tau(\omega, t)).$$

Clearly, for each $t \in \mathbb{R}^d$, these simple random variables converge to $\tau(\omega, t)$ P-a.s. Hence, $X_{\tau_n} \to X_\tau$ P-a.s. for each $t \in \mathbb{R}^d$ because X has continuous sample paths. Since X_{τ_n} is strictly stationary by the proof of Proposition 4.3.5 part (1) for all n (noting that above we did not use the fact that τ has $C^1(\mathbb{R}^d; \mathbb{R}^d)$ realizations and hence the proof still holds), we can argue similarly to how we did in the

proof of Proposition 2.3.4 part (1) to see that X_τ must be strictly stationary. We prove that X and X_τ are jointly strictly stationary in a similar way.

It remains to prove parts (2)-(4). Since X P-a.s. has continuous realizations, without loss of generality we may assume that $X(\omega, t)$ is finite for all $\omega \in \Omega, t \in \mathbb{R}^d$. For $M > 0$, define $f_M : \mathbb{C} \to \mathbb{C}$ by

$$f_M(z) = \begin{cases} z, & |z| \leq M, \\ M\frac{z}{|z|}, & |z| > M, \end{cases}$$

which is Lebesgue measurable. As in the proof of Proposition 2.3.4, $f_M(X)$ converges pointwise to X for all $\omega \in \Omega, t \in \mathbb{R}^d$, as $M \to \infty$. Thus, $f_M(X)_\tau(\omega, t) = f_M(X)(\omega, t - \tau(\omega, t)) \to X(\omega, t - \tau(\omega, t)) = X_\tau(\omega, t)$ as $M \to \infty$. By the Monotone Convergence Theorem [20], since $|f_M(X)_\tau| \leq |f_N(X)_\tau|$ for $M \leq N$, we have that, for each $t \in \mathbb{R}^d$

$$\lim_{M\to\infty} \mathbb{E}\left[|f_M(X)_\tau(\cdot, t)|\right] = \mathbb{E}\left[|X_\tau(\cdot, t)|\right].$$

Given that $|f_M(X)| \leq M$ everywhere, we see that $|f_M(X)_{\tau_n}| \leq M$ everywhere. Also, since f_M is continuous, $f_M(X)$ P-a.s. has continuous sample paths, and so $f_M(X)_{\tau_n} \to f_M(X)_\tau$ P-a.s. for each $t \in \mathbb{R}^d$. Thus, by the Lebesgue dominated convergence theorem [20], we have that

$$\lim_{n\to\infty} \mathbb{E}\left[|f_M(X)_{\tau_n}(\cdot, t)|\right] = \mathbb{E}\left[|f_M(X)_\tau(\cdot, t)|\right] \text{ for all } t \in \mathbb{R}^d$$

But, from Proposition 4.3.5 part (2), $\mathbb{E}\left[|f_M(X)_{\tau_n}(\cdot, t)|\right] = \mathbb{E}\left[|f_M(X)(\cdot, t)|\right]$ for all $t \in \mathbb{R}^d$ and for all n. So, $\mathbb{E}\left[|f_M(X)_\tau(\cdot, t)|\right] = \mathbb{E}\left[|f_M(X)(\cdot, t)|\right]$ for all $t \in \mathbb{R}^d$. Applying the limits above, we thus see

that

$$\mathbb{E}\left[|X_\tau(\cdot,t)|\right] = \lim_{M\to\infty} \mathbb{E}\left[|f_M(X)_\tau(\cdot,t)|\right] = \lim_{M\to\infty} \mathbb{E}\left[|f_M(X)(\cdot,t)|\right] = \mathbb{E}\left[|X(\cdot,t)|\right].$$

Parts (3) and (4) are similarly proven. □

Remark 4.3.8. *Propositions 4.3.5 and 4.3.7 give us conditions under which X_τ is a mean square continuous, strictly stationary process.*

4.3.3.2 Proving Stability

Next, note that in Theorem 2.2.15 on thestability of deterministic scattering transforms to small deformations, the stability estimated was only proved for (ε, R)-bandlimited functions. In order to prove stability estimates for the stationary case, we need a similar definition.

Definition 4.3.9. *Let $R > 0$ and $\varepsilon \in [0,1)$. We say that a strictly stationary process X is (ε, R)-bandlimited if the measure μ_X corresponding to the autocorrelation function R_X satisfies*

$$(1-\varepsilon)\mu_X(\mathbb{R}^d) \leq \mu_X(Q_R(0)). \tag{4.11}$$

There are two things to note. First, if $\varepsilon = 0$, then Equation (4.11) says that μ_X is supported on $Q_R(0)$, analogous to R-bandlimited functions. Second, for fixed $\varepsilon > 0$, for every mean square continuous strictly stationary process X there exists $R > 0$ sufficiently large such that Equation (4.11) holds.

With this, we are ready to prove the following stability result.

Theorem 4.3.10. *Let $X : \Omega \times \mathbb{R}^d \to \mathbb{C}$ be a strictly stationary process which is mean square continuous and (ε, R)-bandlimited. Let $\tau : \Omega \times \mathbb{R}^d \to \mathbb{R}^d$ be a strictly stationary process, inde-*

pendent of X, such that P-a.s. τ is $C^1(\mathbb{R}^d;\mathbb{R}^d)$ in time with $\|\nabla\tau(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)} \leq \frac{1}{2d}$ P-a.s. and $\mathbb{E}[\|\tau(\omega,\cdot)\|^2_{L^\infty(\mathbb{R}^d)}] < \infty$. Further, suppose that X and τ satisfy the conditions of Proposition 4.3.5 or Proposition 4.3.7, and let $X_\tau(\omega,t) \coloneqq X(\omega,t-\tau(\omega,t))$. Then, there exists a constant $C > 0$ (that does not depend on X and τ) such that for all $t \in \mathbb{R}^d$

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X_\tau(\cdot,t)|^2\right] \leq C\left(R^2\mathbb{E}\left[\|\tau(\omega,\cdot)\|^2_{L^\infty(\mathbb{R}^d)}\right] + \varepsilon\right)\mathbb{E}\left[|X(\cdot,t)|^2\right]. \tag{4.12}$$

Proof. First, let's assume the conditions of Proposition 4.3.7, i.e., that P-a.s. the realizations of X are continuous. Let $\varphi \in \mathcal{S}(\mathbb{R}^d)$ such that $\widehat{\varphi}$ is real-valued and even, supported in $Q_2(0)$, is identically 1 on $Q_1(0)$, and satisfies $0 \leq \widehat{\varphi}(\xi) \leq 1$ for all $\xi \in \mathbb{R}^d$. Define $\varphi_R(t) \coloneqq R^d\varphi(Rt)$ and let $X_R(\omega,t) \coloneqq X * \varphi_R(\omega,t)$.

By Proposition 4.3.7, X and X_τ are mean-square continuous, jointly strictly stationary processes. Thus, by Theorem 4.3.1 we have

$$\mathbb{E}\left[|S[\mathcal{P}]X(\cdot,t) - S[\mathcal{P}]X_\tau(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X(\cdot,t) - X_\tau(\cdot,t)|^2\right].$$

By the triangle inequality and the fact that $(a+b+c)^2 \leq 3a^2+3b^2+3c^2$ for $a,b,c \in \mathbb{R}$,

$$\begin{aligned}\mathbb{E}\left[|X(\cdot,t) - X_\tau(\cdot,t)|^2\right] \leq 3\Big(&\mathbb{E}\left[|X(\cdot,t) - X_R(\cdot,t)|^2\right] \\ &+\mathbb{E}\left[|X_R(\cdot,t) - (X_R)_\tau(\cdot,t)|^2\right] + \mathbb{E}\left[|(X_R)_\tau(\cdot,t) - X_\tau(\cdot,t)|^2\right]\Big).\end{aligned}$$

where $(X_R)_\tau(\omega,t) = X_R(\omega,t-\tau(\omega,t))$.

We shall bound the three terms separately. By assumption, since $\widehat{\varphi}(0) = 1$, $\int_{\mathbb{R}^d} \varphi_R(x)\mathrm{d}x = 1$.

Thus,

$$X(\omega,t) - X * \varphi_R(\omega,t) = \int_{\mathbb{R}^d} \left[X(\omega,t) - X(\omega,s)\right] \varphi_R(t-s)\mathrm{d}s.$$

So,

$$\begin{aligned}
&\mathbb{E}\left[|X(\cdot,t) - X_R(\cdot,t)|^2\right] \\
&= \mathbb{E}\left[\left(\int [X(\cdot,t) - X(\cdot,s)\varphi_R(t-s)\mathrm{d}s\right)\overline{\left(\int [X(\cdot,t) - X(\cdot,s')\varphi_R(t-s')\mathrm{d}s'\right)}\right] \\
&= \mathbb{E}\left[\iint [X(\cdot,t) - X(\cdot,s)][\overline{X(\cdot,t) - X(\cdot,s')}]\varphi_R(t-s)\overline{\varphi_R(t-s')}\mathrm{d}s\mathrm{d}s'\right].
\end{aligned}$$

By the Fubini-Tonelli Theorem, this is equal to

$$\begin{aligned}
&\iint \mathbb{E}\left[[X(\cdot,t) - X(\cdot,s)][\overline{X(\cdot,t) - X(\cdot,s')}]\varphi_R(t-s)\overline{\varphi_R(t-s')}\right]\mathrm{d}s\mathrm{d}s' \\
&= \iint \left(R_X(0) - R_X(t-s') - R_X(s-t) + R_X(s-s')\right)\varphi_R(t-s)\overline{\varphi_R(t-s')}\mathrm{d}s\mathrm{d}s',
\end{aligned}$$

where $R_X(s) = \mathbb{E}\left[X(\cdot,s+r)\overline{X(\cdot,r)}\right] = \int e^{2\pi i\xi\cdot s}\mathrm{d}\mu_X(\xi)$. Notice that

$$\int R_X(s)\varphi_R(s)\mathrm{d}s = \iint e^{2\pi i\xi\cdot s}\mathrm{d}\mu_X(\xi)\varphi_R(s)\mathrm{d}s = \int \widehat{\varphi_R}(\xi)\mathrm{d}\mu_X(\xi),$$

since $\widehat{\varphi_R}$ is even. Since we assumed $\widehat{\varphi_R}$ is real-valued and even, we have that φ_R is real-valued and even. Thus, we have

$$\int R_X(t-s')\overline{\varphi_R(t-s')}\mathrm{d}s' = \int R_X(s-t)\varphi_R(t-s)\mathrm{d}s = \int \widehat{\varphi_R}(\xi)\mathrm{d}\mu_X(\xi).$$

Finally, notice that

$$\iint R_X(s-s')\varphi_R(t-s)\overline{\varphi_R(t-s')}\mathrm{d}s\mathrm{d}s' = \iint R_X(s'-s)\varphi_R(s)\varphi_R(s')\mathrm{d}s\mathrm{d}s'$$
$$= \iiint e^{2\pi i\xi\cdot(s'-s)}\varphi_R(s)\varphi_R(s')\mathrm{d}\mu_X(\xi)\mathrm{d}s\mathrm{d}s' = \int \left(\widehat{\varphi_R}(\xi)\right)^2 \mathrm{d}\mu_X(\xi)$$

since $\widehat{\varphi_R}$ is even and real-valued. Since $\int\limits_{\mathbb{R}^d} \varphi_R(x)\mathrm{d}x = 1$, we thus get

$$\mathbb{E}\left[|X(\cdot,t) - X_R(\cdot,t)|^2\right] = \int |1-\widehat{\varphi_R}(\xi)|^2 \,\mathrm{d}\mu_X(\xi). \tag{4.13}$$

Since $\widehat{\varphi_R} \equiv 1$ on $Q_R(0)$ and since $|1-\widehat{\varphi_R}(\xi)| \leq 1$ for all $\xi \in \mathbb{R}^d$, this is bounded above by

$$\int\limits_{\mathbb{R}^d\setminus Q_R(0)} \mathrm{d}\mu_X(\xi) = \mu_X(\mathbb{R}^d \setminus Q_R(0)) \leq \varepsilon\mu_X(\mathbb{R}^d) = \varepsilon\mathbb{E}\left[|X(\cdot,t)|^2\right]. \tag{4.14}$$

This takes care of the first term.

For the third term, by Proposition 2.3.4 part (2), X and X_R are jointly strictly stationary. Thus, $X - X_R$ is strictly stationary; also, since φ_R is Schwartz, we have by Proposition 2.3.4 part (3) that X_R P-a.s. has continuous realizations. Since this is also true for X, we see that $X - X_R$ P-a.s. has continuous realizations. So, by Proposition 4.3.7 part (3) we have

$$\mathbb{E}\left[|(X_R)_\tau(\cdot,t) - X_\tau(\cdot,t)|^2\right] = \mathbb{E}\left[|(X_R - X)_\tau(\cdot,t)|^2\right] = \mathbb{E}\left[|X_R(\cdot,t) - X(\cdot,t)|^2\right],$$

where $(X_R)_\tau(\omega,t) = X_R(\omega, t-\tau(\omega,t))$. By Equations (4.13) and (4.14), we thus get

$$\mathbb{E}\left[|(X_R)_\tau(\cdot,t) - X_\tau(\cdot,t)|^2\right] \leq \varepsilon\mathbb{E}\left[|X(\cdot,t)|^2\right].$$

This takes care of the third term.

For the second term, notice that

$$(X_R)_\tau(\omega,t) - X_R(\omega,t) = \int_{\mathbb{R}^d} X(\omega,s)(\varphi_R(t-\tau(\omega,t)-s) - \varphi_R(t-s))\mathrm{d}s.$$

Thus, $(X_R)_\tau(\omega,t) - X_R(\omega,t) = K_\tau(\omega)X(\omega,t)$ where $K_\tau(\omega)$ is an integral operator with random kernel $k_\tau(\omega,t,s) = \varphi_R(t-\tau(\omega,t)-s) - \varphi_R(t-s)$, independent of X. We shall show that there exists a function $\widetilde{k}_\tau : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$ such that

$$\mathbb{E}\left[k_\tau(\cdot,t,s)\overline{k_\tau(\cdot,t,s')}\right] = \widetilde{k}_\tau(t-s,t-s'), \tag{4.15}$$

and

$$\iint |\widetilde{k}_\tau(v,v')||v-v'|\mathrm{d}v\mathrm{d}v' < \infty. \tag{4.16}$$

Then, by Lemma 4.8 from [88], we will have that

$$\mathbb{E}\left[|K_\tau(\cdot)X(\cdot,t)|^2\right] \leq \mathbb{E}\left[\|K_\tau(\cdot)\|^2\right]\mathbb{E}\left[|X(\cdot,t)|^2\right],$$

where $\|K_\tau(\cdot)\|$ is the operator norm of a realization of $K_\tau(\omega)$. To prove Equation 4.15, note that

$$\begin{aligned} k_\tau(\omega,t,s)\overline{k_\tau(\omega,t,s')} &= \varphi_R(t-s)\overline{\varphi_R(t-s')} - \varphi_R(t-\tau(\omega,t)-s)\overline{\varphi_R(t-s')} \\ &-\varphi_R(t-s)\overline{\varphi_R(t-\tau(\omega,t)-s')} + \varphi_R(t-\tau(\omega,t)-s)\overline{\varphi_R(t-\tau(\omega,t)-s')}. \end{aligned}$$

We take the expectation of each term in this sum. Clearly, $\mathbb{E}\left[\varphi_R(t-s)\overline{\varphi_R(t-s')}\right] = \varphi_R(t-$

$s)\overline{\varphi_R(t-s')}$ only depends on $t-s$ and $t-s'$. For the second term, we use the strict stationarity of τ to see that

$$\begin{aligned}\mathbb{E}\left[\varphi_R(t-\tau(\cdot,t)-s)\overline{\varphi_R(t-s')}\right] &= \overline{\varphi_R(t-s')}\mathbb{E}\left[\varphi_R(t-\tau(\cdot,t)-s)\right] \\ &= \overline{\varphi_R(t-s')}\mathbb{E}\left[\varphi_R(t-s-\tau(\cdot,t-s))\right].\end{aligned}$$

Thus, this term only depends on τ, $t-s$, and $t-s'$; similarly,

$$\mathbb{E}\left[\varphi_R(t-s)\overline{\varphi_R(t-\tau(\cdot,t)-s')}\right] = \varphi_R(t-s)\mathbb{E}\left[\overline{\varphi_R(t-s'-\tau(\cdot,t-s'))}\right].$$

For the last term, notice that $s-s' = (t-s')-(t-s)$. Hence, once again using strict stationarity, we get

$$\begin{aligned}&\mathbb{E}\left[\varphi_R(t-\tau(\cdot,t)-s)\overline{\varphi_R(t-\tau(\cdot,t)-s')}\right] \\ =\,&\mathbb{E}\left[\varphi_R(t-\tau(\cdot,s-s')-s)\overline{\varphi_R(t-\tau(\cdot,s-s')-s')}\right].\end{aligned}$$

Adding up the terms, we see that $\mathbb{E}\left[k_\tau(\cdot,t,s)\overline{k_\tau(\cdot,t,s')}\right]$ only depends on τ, $t-s$, and $t-s'$, proving (4.15).

By the Fubini-Tonelli Theorem

$$
\begin{aligned}
&\iint |\widetilde{k}(v,v')||v-v'|\mathrm{d}v\mathrm{d}v' \\
&\leq \iint \mathbb{E}\left[\left|k_\tau(\cdot,0,-v)\overline{k_\tau(\cdot,0,-v')}\right|\right]|v-v'|\mathrm{d}v\mathrm{d}v' \\
&= \mathbb{E}\left(\iint |k_\tau(\cdot,0,-v)k_\tau(\cdot,0,-v')||v-v'|\mathrm{d}v\mathrm{d}v'\right) \\
&\leq \mathbb{E}\left(\iint |\varphi_R(v)\varphi_R(v')||v-v'|\mathrm{d}v\mathrm{d}v'\right) \\
&\quad + \mathbb{E}\left(\iint |\varphi_R(v-\tau(\cdot,0))\varphi_R(v')||v-v'|\mathrm{d}v\mathrm{d}v'\right) \\
&\quad + \mathbb{E}\left(\iint |\varphi_R(v)\varphi_R(v'-\tau(\cdot,0))||v-v'|\mathrm{d}v\mathrm{d}v'\right) \\
&\quad + \mathbb{E}\left(\iint |\varphi_R(v-\tau(\cdot,0))\varphi_R(v'-\tau(\cdot,0))||v-v'|\mathrm{d}v\mathrm{d}v'\right).
\end{aligned}
$$

We shall show that each of these terms is finite. Since $\varphi_R \in \mathcal{S}(\mathbb{R}^d)$, it is clear that the first term is finite. For the last term, after the change of variables $v \mapsto v+\tau(\omega,0)$ and $v' \mapsto v'+\tau(\omega,0)$, note that

$$
\iint |\varphi_R(v-\tau(\omega,0))\varphi_R(v'-\tau(\omega,0))||v-v'|\mathrm{d}v\mathrm{d}v' = \iint |\varphi_R(v)\varphi_R(v')||v-v'|\mathrm{d}v\mathrm{d}v'
$$

is finite and independent of ω. Taking expectation, the last term is also finite. For the second term, notice that, by a change of variables,

$$
\iint |\varphi_R(v-\tau(\omega,0))\varphi_R(v')||v-v'|\mathrm{d}v\mathrm{d}v' = \iint |\varphi_R(v)\varphi_R(v')||v+\tau(\omega,0)-v'|\mathrm{d}v\mathrm{d}v'.
$$

Since $|\tau(\omega,0)| \leq \|\tau(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)}$ and $\mathbb{E}\left[\|\tau(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)}\right] \leq \left(\mathbb{E}\left[\|\tau(\omega,\cdot)\|^2_{L^\infty(\mathbb{R}^d)}\right]\right)^{1/2} < \infty$ by Jensen's inequality, it's easily seen that the expectation of this integral is finite. Thus, the sec-

ond term is finite, and similarly so is the third term. Combining, these results prove (4.16). Hence, $\mathbb{E}\left[|K_\tau(\cdot)X(\cdot,t)|^2\right] \leq \mathbb{E}\left[\|K_\tau(\cdot)\|^2\right]\mathbb{E}\left[|X(\cdot,t)|^2\right]$.

It remains to bound $\mathbb{E}\left[\|K_\tau(\cdot)\|^2\right]$. Since we assumed that $\tau(\omega,\cdot)$ is P-a.s. $C^1(\mathbb{R}^d;\mathbb{R}^d)$, by the proof of Theorem 3.6(d) from [44],

$$\|K_\tau(\omega)\|^2 \leq 4R^2\|\nabla\varphi\|_{L^1}^2\|\tau(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)}^2 \; P\text{-a.s.}$$

Thus,

$$\mathbb{E}\left[\|K_\tau(\cdot)\|^2\right] \leq 4R^2\|\nabla\varphi\|_{L^1}^2\mathbb{E}\left[\|\tau(\omega,\cdot)\|_{L^\infty(\mathbb{R}^d)}^2\right].$$

Taking $C = 3\max\{2, 4\|\nabla\varphi\|_{L^1}^2\}$ completes the proof.

Now, suppose the assumptions of Proposition 4.3.5, i.e., that τ takes countably many values. We still have that X and X_R are jointly strictly stationary by Proposition 2.3.4 part (2), and so $(X_R)_\tau$, X_τ are jointly strictly stationary by Proposition 4.3.5. Thus, by the same Proposition, we have that

$$\mathbb{E}\left[|(X_R)_\tau(\cdot,t) - X_\tau(\cdot,t)|^2\right] \leq \mathbb{E}\left[|X_R(\cdot,t) - X(\cdot,t)|^2\right] \leq \varepsilon\mathbb{E}\left[|X(\cdot,t)|^2\right].$$

The rest of the proof remains the same. □

We have proved here several general properties of stationary scattering transforms, but there are many other properties one might desire, such as fast decay of the energy of layers. In Section 4.4, we detail one particular kind of stationary scattering transforms that exhibits such negligibility of higher-order outputs.

4.4 Energy Decay for Stationary Fourier Scattering Transforms

In this section, we focus on a special case of the stationary scattering transform called the *stationary Fourier scattering transform*, following inspiration from the deterministic FST constructed by Czaja and Li [44, 45]. We shall prove that the energy over paths $p \in \Lambda_F^K$ of the stationary Fourier scattering transform decays exponentially fast in K, and hence that higher-order terms are negligible.

4.4.1 Definitions and Setup

First, let Λ_F be a countable indexing set with $0 \notin \Lambda_F$, and let $\mathcal{G}_F = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_F} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a uniform covering frame (see Definition 2.2.8). Let $\mathcal{P}_F = \bigcup_{k=0}^{\infty} \Lambda_F^k$ be the set of paths. With this, we can define the stationary Fourier scattering transform (cf. Definition 2.2.10).

Definition 4.4.1. *A stationary Fourier scattering transform $S[\mathcal{P}_F]$ of a mean square continuous, strictly stationary process X is defined to be a stationary scattering transform (see Definition 4.2.2) of X whose semi-discrete Bessel sequence $\mathcal{G}_F$ is a uniform covering frame and whose stationary scattering propagators are $U_F[p]$ for $p \in \mathcal{P}_F$.*

As a stationary Fourier scattering transform is by definition a stationary scattering transform, it inherits all of the properties proved above (namely, the properties of Theorems 4.3.1, 4.3.3, and 4.3.10, and Corollary 4.3.2). Additionally, we shall show that, for all mean square continuous, strictly stationary processes X, the energy over paths of length $p \in \Lambda_F^K$ for fixed K of the stationary Fourier scattering propagator, i.e.,

$$\sum_{p \in \Lambda_F^K} \mathbb{E}\left[|U_F[p]X(\cdot, t)|^2\right]$$

decays exponentially fast in K (see Theorem 4.4.3, and cf. Theorem 2.2.18). As a consequence, we

shall also show that the stationary Fourier scattering transform preserves energy, i.e.,

$$\mathbb{E}\left[|X(\cdot,t)|^2\right] = \mathbb{E}\left[|S[\mathcal{P}_{\mathcal{F}}]X(\cdot,t)|^2\right], \text{ for all } t \in \mathbb{R}^d.$$

In order to prove the results on energy decay and preservation, we need the following result, cf. Proposition 3.1 in [44].

Lemma 4.4.2. *Let X be a strictly stationary process which is mean square continuous. Then, for a stationary Fourier scattering transform $S[\mathcal{P}_F]$ with uniform covering frame $\mathcal{G}_F$ and for every $k \geq 0$,*

$$\sum_{p\in\Lambda_F^k} \mathbb{E}\left[|U_F[p]X * g_0(\cdot,t)|^2\right] + \sum_{p'\in\Lambda_F^{k+1}} \mathbb{E}\left[|U_F[p']X(\cdot,t)|^2\right] = \sum_{p\in\Lambda_F^k} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right], \tag{4.17}$$

for all $t \in \mathbb{R}^d$. Also, for every $K \in \mathbb{N}$

$$\sum_{p\in\Lambda_F^{K+1}} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] + \sum_{k=0}^{K} \sum_{p\in\Lambda_F^k} \mathbb{E}\left[|U_F[p]X * g_0(\cdot,t)|^2\right] = \mathbb{E}\left[|X(\cdot,t)|^2\right], \tag{4.18}$$

for all $t \in \mathbb{R}^d$.

Proof. For each path $p \in \Lambda^k$, let $\mu_{F,p}$ be the bounded Borel measure such that $\mathbb{E}\left[|U_F[p]X|^2(\cdot,t)\right] = \int_{\mathbb{R}^d} 1 d\mu_{F,p}$. By the frame condition and Proposition 2.3.3 part (5), notice that

$$\begin{aligned}
&\mathbb{E}\left[|U_F[p]X * g_0(\cdot,t)|^2\right] + \sum_{\lambda\in\Lambda_F} \mathbb{E}\left[|U_F[p+\lambda]X(\cdot,t)|^2\right] \\
&= \int_{\mathbb{R}^d} |\widehat{g_0}|^2 + \sum_{\lambda\in\Lambda_F} |\widehat{g_\lambda}|^2 \mathrm{d}\mu_{F,p} = \int_{\mathbb{R}^d} 1 \mathrm{d}\mu_{F,p} = \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right].
\end{aligned}$$

Summing over $p \in \Lambda_F^k$ and noting that every $p' \in \Lambda_F^{k+1}$ can be written as $p' = (p,\lambda)$ for $\lambda \in \Lambda_F$, we

see for all $t \in \mathbb{R}^d$ that

$$\sum_{p \in \Lambda_F^k} \mathbb{E}\left[|U_F[p]X * g_0(\cdot, t)|^2\right] + \sum_{p' \in \Lambda_F^{k+1}} \mathbb{E}\left[|U_F[p']X(\cdot, t)|^2\right] = \sum_{p \in \Lambda_F^k} \mathbb{E}\left[|U_F[p]X(\cdot, t)|^2\right],$$

which proves Equation (4.17). Summing both sides from $k = 0$ to K, using $\Lambda_F^0 = \{\varnothing\}$ with $U_F[\varnothing]X = X$, and subtracting $\sum_{k=1}^{K} \sum_{p \in \Lambda_F^k} \mathbb{E}\left[|U_F[p]X(\cdot, t)|^2\right]$ from both sides, we get Equation (4.18).

□

4.4.2 Energy Decay Proof

Now, we shall show that the energy over paths $p \in \Lambda_F^K$ of the stationary Fourier scattering propagator decays exponentially in K (cf. Theorem 2.2.18). As a consequence, we shall see that the stationary Fourier scattering transform is energy preserving (see Corollary 4.4.4).

Theorem 4.4.3. *Let $S[\mathcal{P}_F]$ be a stationary Fourier scattering transform with corresponding uniform covering frame $\mathcal{G}_F$ and stationary scattering propagators $U_F[p]$ for $p \in \mathcal{P}_F$. There exists a constant $C_0 \in (0, 1)$ (depending only on the uniform covering frame) such that for every strictly stationary process X which is mean square continuous and every integer $K \geq 1$,*

$$C_0^{K-1}\mathbb{E}\left[|X * g_0(\cdot, t)|^2\right] + \sum_{p \in \Lambda_F^K} \mathbb{E}[|U_F[p]X(\cdot, t)|^2] \leq C_0^{K-1}\mathbb{E}\left[|X(\cdot, t)|^2\right], \text{ for all } t \in \mathbb{R}^d.$$

Proof. We begin by following a similar argument in the proof of Proposition 3.3 in [44], and note that there exists $\varphi \in L^1 \cap L^2(\mathbb{R}^d)$ such that $\varphi \geq 0$, $\widehat{\varphi}$ is continuous and is decreasing along each Euclidean axis, $|\widehat{\varphi}(0)| > 0$, and $|\widehat{\varphi}(\xi)| \leq |\widehat{g_0}(\xi)|$ for all $\xi \in \mathbb{R}^d$. There exists $R = R_\varphi > 0$ and $C = C_\varphi \in (0, 1)$

such that $|\widehat{\varphi}(\xi)|^2 \geq C$ for all $\xi \in Q_R(0)$. By the uniform covering property, there exists $N = N_R$ such that for for all $\lambda \in \Lambda_F$, there exists $\{\xi_{\lambda,n} : n = 1, \ldots, N\} \subseteq \mathbb{R}^d$ such that

$$\text{supp}(\widehat{g_\lambda}(\xi)) \subseteq \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n}).$$

Let $1 \leq k \leq K$ and fix $q \in \Lambda_F^{k-1}$. From the setup and Proposition 2.3.3 Part (5), we see that there exists a bounded Borel measure $\mu_{F,q}$ such that

$$\mathbb{E}\left[|U_F[q]X * g_\lambda(\cdot, t)|^2\right] = \int_{\mathbb{R}^d} |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\mu_{F,q}(\xi)$$

where the right-hand side is independent of t by stationarity. Applying the uniform covering property above and the fact that $C \leq |\widehat{\varphi}(\xi - \xi_{\lambda,n})|$ for $\xi \in Q_R(\xi_{\lambda,n})$, we see that

$$\begin{aligned}
\int_{\mathbb{R}^d} |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\mu_{F,q}(\xi) &\leq \sum_{n=1}^{N} \int_{Q_R(\xi_{\lambda,n})} |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\mu_{F,q}(\xi) \\
&\leq \frac{1}{C} \sum_{n=1}^{N} \int_{Q_R(\xi_{\lambda,n})} |\widehat{g_\lambda}(\xi)\widehat{\varphi}(\xi - \xi_{\lambda,n})|^2 \mathrm{d}\mu_{F,q}(\xi) \\
&\leq \frac{1}{C} \sum_{n=1}^{N} \int_{\mathbb{R}^d} |\widehat{g_\lambda}(\xi)\widehat{\varphi}(\xi - \xi_{\lambda,n})|^2 \mathrm{d}\mu_{F,q}(\xi) \\
&= \frac{1}{C} \sum_{n=1}^{N} \mathbb{E}\left[|U_F[q]X * g_\lambda * M_{\xi_{\lambda,n}}\varphi(\cdot, t)|^2\right].
\end{aligned}$$

For each $t \in \mathbb{R}^d$, it is clear that $|U_F[q]X * g_\lambda * M_{\xi_{\lambda,n}}\varphi(\omega, t)| \leq |U_F[q]X * g_\lambda| * |M_{\xi_{\lambda,n}}\varphi|(\omega, t) =$

$|U_F[q]X * g_\lambda| * \varphi(\omega, t)$ P-a.s. since φ is real-valued and non-negative. Hence,

$$\frac{1}{C}\sum_{n=1}^{N}\mathbb{E}\left[|U_F[q]X * g_\lambda * M_{\xi_{\lambda,n}}\varphi(\cdot,t)|^2\right] \leq \frac{1}{C}\sum_{n=1}^{N}\mathbb{E}\left[\left||U_F[q]X * g_\lambda| * \varphi(\cdot,t)\right|^2\right]$$
$$= \frac{N}{C}\mathbb{E}\left[|U_F[(q,\lambda)]X * \varphi(\cdot,t)|^2\right].$$

The last equality holds since the interior of the sum on the right-hand side of the inequality is independent of N and $U_F[q]X * g_\lambda = U_F[(q,\lambda)]X$. Note that

$$\mathbb{E}\left[|U_F[(q,\lambda)]X * \varphi(\cdot,t)|^2\right] = \int_{\mathbb{R}^d} |\widehat{\varphi}(\xi)|^2 \mathrm{d}\mu_{F,(q,\lambda)}(\xi)$$
$$\leq \int_{\mathbb{R}^d} |\widehat{g_0}(\xi)|^2 \mathrm{d}\mu_{F,(q,\lambda)}(\xi) = \mathbb{E}\left[|U_F[(q,\lambda)]X * g_0(\cdot,t)|^2\right].$$

Thus, we have that

$$\mathbb{E}\left[|U_F[q]X * g_\lambda(\cdot,t)|^2\right] \leq \frac{N}{C}\mathbb{E}\left[||U_F[q]X * g_\lambda| * g_0(\cdot,t)|^2\right].$$

Note that N, C are independent of $q \in \Lambda_F^{k-1}$ and $\lambda \in \Lambda_F$. Thus, summing over λ, q, and noting that every path $p \in \Lambda_F^k$ can be written as $p = (q,\lambda)$, we get that

$$\frac{C}{N}\sum_{p\in\Lambda_F^k}\mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] \leq \sum_{p\in\Lambda_F^k}\mathbb{E}\left[||U_F[p]X| * g_0(\cdot,t)|^2\right].$$

Let $C_0 = 1 - \frac{C}{N}$. Applying Equation (4.17) to the right-hand side and rearranging, we get that

$$\sum_{p\in\Lambda_F^{k+1}}\mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] \leq C_0\sum_{p\in\Lambda_F^k}\mathbb{E}\left[|U_F[p]X|(\cdot,t)|^2\right].$$

Iterating this inequality, and applying Equation (4.17) with $k = 0$, we see that

$$\sum_{p \in \Lambda_F^K} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] \leq C_0^{K-1} \sum_{p \in \Lambda_F} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right]$$

$$= C_0^{K-1}\left(\mathbb{E}\left[|X(\cdot,t)|^2\right] - \mathbb{E}\left[|X * g_0(\cdot,t)|^2\right]\right)$$

□

From this result, we can immediately show that the stationary Fourier scattering transform is energy preserving (cf. Corollary 2.2.19).

Corollary 4.4.4. *Let $S[\mathcal{P}_F]$ be a stationary Fourier scattering transform, and let X be a strictly stationary process which is mean square continuous. Then,*

$$\mathbb{E}\left[|X(\cdot,t)|^2\right] = \mathbb{E}\left[|S[\mathcal{P}_\mathcal{F}]X(\cdot,t)|^2\right], \textit{ for all } t \in \mathbb{R}^d. \tag{4.19}$$

Proof. By Lemma 4.18, we know that for all $K \in \mathbb{N}$

$$\sum_{p \in \Lambda_F^{K+1}} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] + \sum_{k=0}^{K} \sum_{p \in \Lambda_F^k} \mathbb{E}\left[|U_F[p]X * g_0(\cdot,t)|^2\right] = \mathbb{E}\left[|X(\cdot,t)|^2\right]. \tag{4.20}$$

Thus,

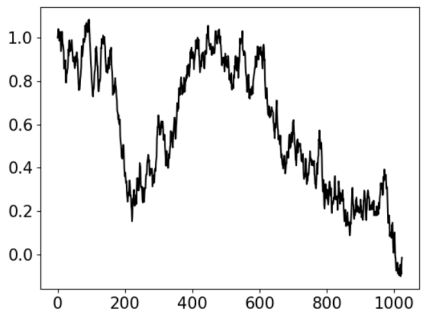
$$\mathbb{E}\left[|S[\mathcal{P}_\mathcal{F}]X(\cdot,t)|^2\right] = \lim_{K\to\infty} \sum_{k=0}^{K} \sum_{p \in \Lambda_F^k} \mathbb{E}\left[|U_F[p]X * g_0(\cdot,t)|^2\right]$$

$$= \mathbb{E}\left[|X(\cdot,t)|^2\right] - \lim_{K\to\infty} \sum_{p \in \Lambda_F^{K+1}} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] = \mathbb{E}\left[|X(\cdot,t)|^2\right],$$

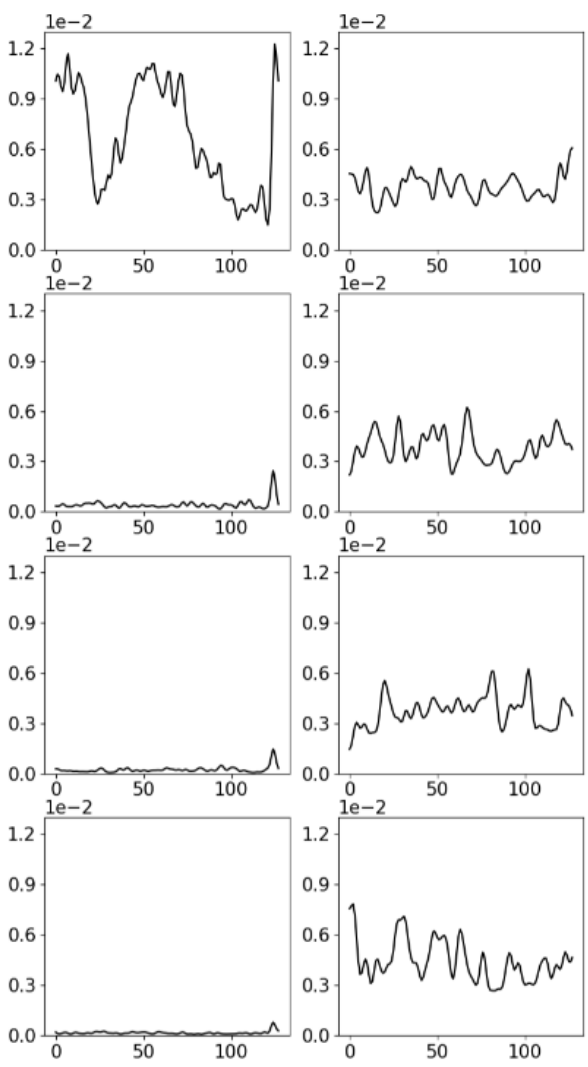
since $\lim_{K\to\infty} \sum_{p\in\Lambda_F^{K+1}} \mathbb{E}\left[|U_F[p]X(\cdot,t)|^2\right] = 0$ by Theorem 4.4.3. □

4.4.3 Comparing Stationary Wavelet and Fourier Scattering Features

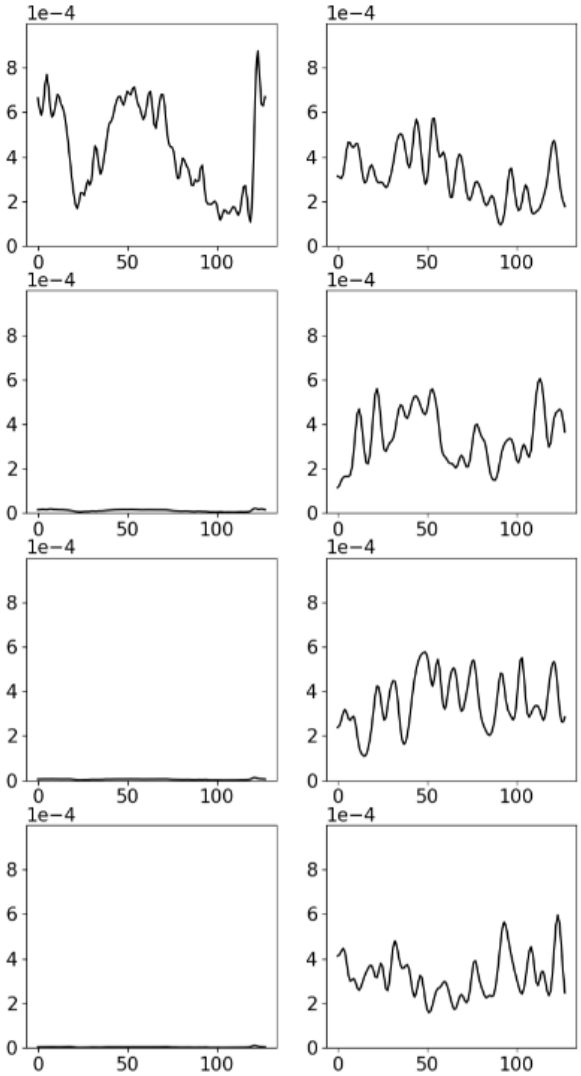
We close this chapter with a case study to test the suitability of the proposed stationary scattering transform. Our definition enables us to use different features constructions that can be adapted to specific processes. In Figure 4.1, we provide a visual comparison of the features that can be extracted by different scattering transforms. The top image is an instance of a Ornstein-Uhlenbeck process (see [74] for the definition). The bottom images display the outputs of the first (i.e., one filter is applied) and second (i.e., two filters) layers of the Fourier and wavelet scattering transforms. The scattering transform was calculated using the package Kymatio [13], with 3 scales and 6 wavelets per octave [12]. The Fourier scattering transform was calculated using code from `https://github.com/weilinlimath/1D-Fourier-Scattering-Transform`, with a window size of 32, frequency density of 0.5, and circular convolution.



(a) Instance of an Ornstein-Uhlenbeck process.



(b) First 4 layer 1 scattering outputs (Fourier on the left, wavelet on the right)



(c) First 4 layer 2 scattering outputs (Fourier on the left, wavelet on the right)

Figure 4.1: A comparison of scattering transform features of a strictly stationary process. In subfigure 4.1a, we have an instance of an Ornstein-Uhlenbeck process. In subfigures 4.1b and 4.1c, we have 4 first and second layer Fourier and wavelet scattering outputs of the instance.

Chapter 5: Hyperscat: Applications to Hyperspectral Reconstruction

5.1 Introduction

The hyperspectral reconstruction problem, as detailed in Section 2.5.2, consists of generating HSI from corresponding RGB images or MSI which have far fewer channels than the desired HSI. The difficulty in this problem arises when trying to find a map from a low-dimensional to a high-dimensional space, as this is in general ill-posed. Hence, we turn to ideas from data fusion and modality adaptation. Rather than attempting to find an algorithm that directly matches the RGB images or MSI to the corresponding HSI, we instead first embed both modalities into a common feature space. Then, we attempt to find an algorithm for matching these embeddings; as long as the embedding is invertible, we then obtain a map between the modalities; see Figure 5.1.

A variety of modality adaptation and data fusions models, for different applications, have been devised that fit into the schema illustrated in Figure 5.1. For example, diffusion maps were used in [38] as the embedding and rotation-based methods were used for matching, with applications to detecting changes in HSI. Similarly, Laplacian eigenmap embeddings and rotation-based matching have been used in [33–37, 49] and were applied to problems of noise reduction of HSI as well as fusing HSI with LIDAR data. Laplacian eigenmaps were also used in [39, 51] as embeddings, in concert with graph-matching algorithms, for superresolution problems for Magnetic Resonance Imaging (MRI).

In this chapter, we detail a novel modality adaptation scheme called *Hyperscat* for HSI recon-

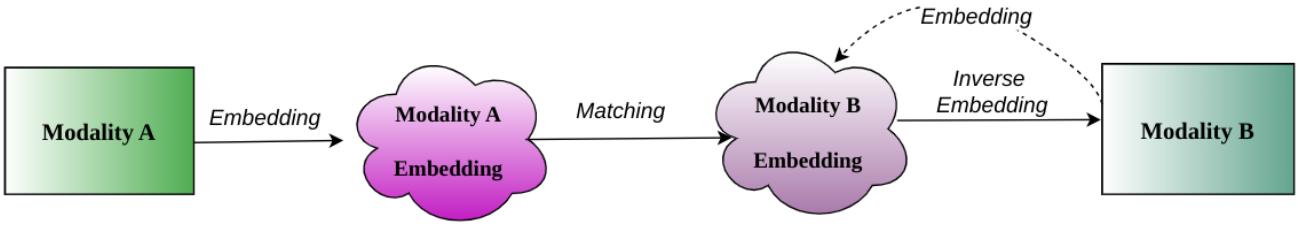


Figure 5.1: The general framework for modality adaptation. Modality A is embedded into a feature space, and then the features of modality A are matched to the features of modality B. Finally, the embedding is inverted to obtain modality B.

struction based on the matching of scattering transform features. More specifically, we calculate a discretized scattering transform of the input RGB images or MSIs. Then, a shallow CNN is trained to match the scattering coefficients of these input images to the scattering coefficients of the corresponding HSI. Finally, we apply an approximate inverse to invert the scattering embedding and obtain the HSI. This pipeline not only achieves comparable results to more complicated machine learning-based methods but is also more computationally efficient, hence making it more viable to train the networks when fewer computational resources are available.

In this chapter, we provide details on our pipeline which uses a discretized WST as the embedding. This pipeline achieved a top 5 numerical result at the 2024 ICASSP Grand Challenge on Hyperspectral Skin Vision, and we discuss those results as well as a variety of reconstruction experiments on the Hyper-Skin and ARAD datasets, including from the many experiments we conducted when attempting to optimize our pipeline. We also briefly consider methods which reduce the computational resources used without significantly degrading performance. This chapter is based on work I conducted with Wojciech Czaja, Jeremiah Emidih, and Richard G. Spencer from [40, 41].

5.2 Hyperscat Pipeline Architecture

Our pipeline consists of three main components. First, the scattering transform is applied to the RGB images or MSI. Then, these scattering coefficients are reshaped and matched by a CNN to the corresponding HSI scattering coefficients. Finally, an inverse DCGAN is applied to invert the HSI scattering embedding and obtain the reconstructed HSI. Here, we discuss these three aspects of our pipeline. We also discuss how these networks can be split into multiple pieces that each reconstruct only part of the hyperspectral image, in order to further reduce computational costs, and we detail an optional network applied at the end of the process which attempts to further improve the reconstruction in this case. Note that, while we will consider modifications of these networks during our optimization experiments (see Section 5.5), the networks we present in this section are the "optimal" versions and what we use when comparing against the baseline network (see Section 5.6).

5.2.1 Wavelet Scattering Transform

We use the implementation of the discretized WST constructed in [13] and detailed in Section 2.4.1. In our experiments, we choose to use $J = 2$ dilations and $L = 4$ rotations. We will apply the WST to images of size $\widehat{C} \times M \times N$ where M, N are the spatial dimensions and $\widehat{C}$ is the number of channels; the 2d convolutions are applied to each channel individually. If the image is RGB (respectively, MSI), then $\widehat{C} = 3$ (respectively, $\widehat{C} = 4$). If the image is HSI, then $\widehat{C} \in \{31, 61\}$, depending on the number of hyperspectral channels measured in the dataset. The scattering output is thus of size $\left(\widehat{C}, S, \frac{M}{2^2}, \frac{N}{2^2}\right)$, where $S = 25$ is the number of scattering channels. A diagram of the discretized WST applied to an MSI can be seen in Figure 5.2.

Figure 5.3 displays the scattering coefficients (from the first and second layers) of a highlighted

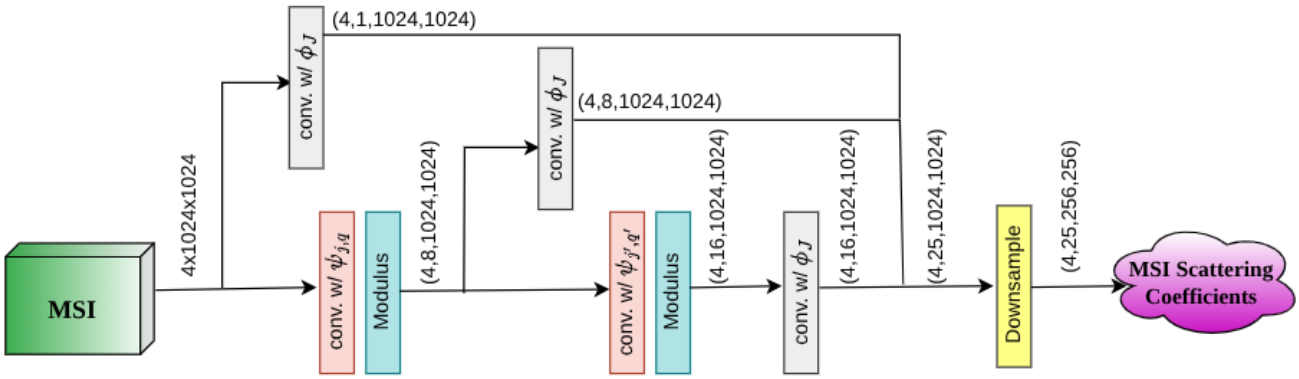
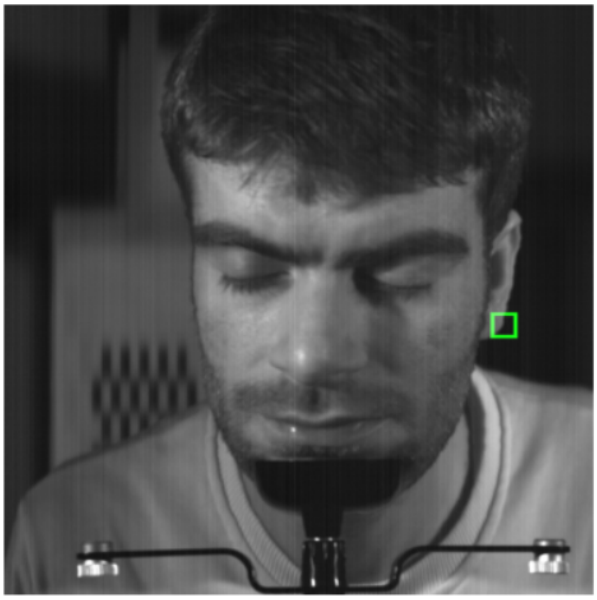


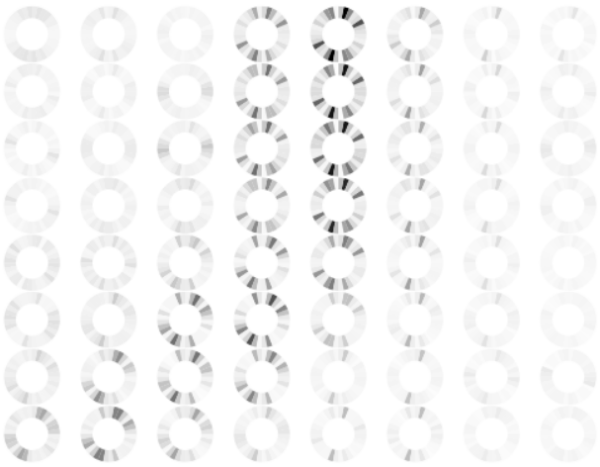
Figure 5.2: A diagram of the discretized WST with $J = 2$ dilations and $Q = 4$ rotations. Each layer involves convolving with some $\psi_{j,q}$ followed by complex modulus. The output of each layer is convolved with ϕ_J, and these outputs are then collected together as the scattering coefficients. In this diagram, the WST is applied to an example MSI of size $4 \times 1024 \times 1024$ and outputs scattering coefficients of size $(4, 25, 256, 256)$.

portion of a channel of an HSI from the Hyper-Skin dataset. Each annulus corresponds to the scattering transform outputs downsampled from the $2^2 \times 2^2$ square of pixels in the highlighted area. The left group of annuli represent the scattering coefficients from the first layer, while the right group represents the second layer outputs. Each annulus is divided into pairs of opposite sectors, with each pair of sectors being the value obtained by applying a filter(s) with a particular combination(s) of rotations and scales. These values are represented by colors which correspond to the relative size of the scattering coefficients in that layer, from white, which represents the smallest relative value, to black, which represents the largest relative value.

For the first layer coefficients, the angle of the sector pairs represents the corresponding rotation which was used by the convolutional filter. The location of the sector pairs corresponds to which scale was used; the outer layer corresponds to $j = 1$, while the inner layer corresponds to $j = 2$. For the second layer coefficients, each pair of sectors corresponding to a particular angle from the first layer is subdivided into 4 subsector pairs, each corresponding to the rotation used by the filter in the second



(a) First layer scattering coefficients (*one filter applied*).



(b) Second layer scattering coefficients (*two filters applied*).

Figure 5.3: One channel of a hyperspectral image (above) and the scattering coefficients (bottom left and bottom right) of the green highlighted region. In (a), each sector corresponds to a rotation and dilation ($j = 1$ for the outer layer, $j = 2$ for the inner layer). In (b), each rotation in (a) is split into 4 subsectors, each corresponding to a second rotation. The colors correspond to the (relative) size of the scattering coefficients in that layer, from smallest (white) to largest (black).

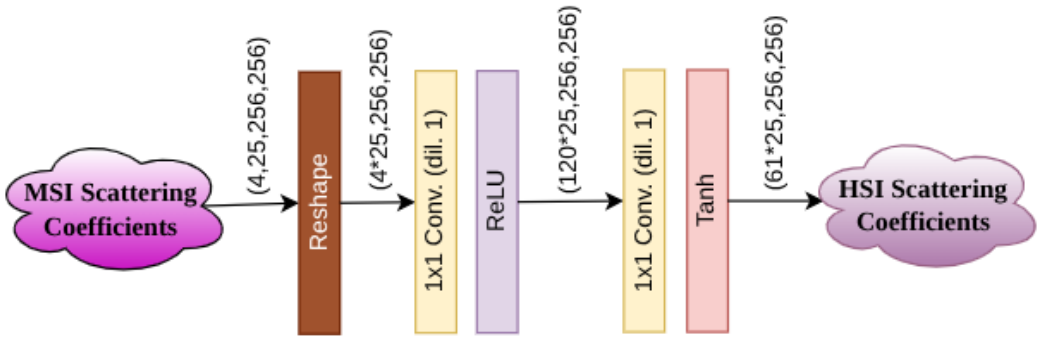


Figure 5.4: A diagram of the NDT (No-Dilation-Together) matching network architecture. The input is the scattering coefficients of an MSI, and are of size $(4, 25, 256, 256)$, where the scattering transform is the one described in Section 5.2.1. The network consists of two convolutional layers (with bias) with window size 1×1, and the output consists of the corresponding HSI scattering coefficients.

layer.

As can be seen in Figure 5.3, the discretized WST appears to detect edges. In the black space behind the subject in the image or in the subject's ear itself, the color is approximately constant, and so the scattering coefficients are small. However, on the edge of the ear, the color changes rapidly, leading to large scattering coefficients. Moreover, note that as the direction of the ear changes, the relatively largest scattering coefficients occur for the angle that faces the same direction as the edge of the ear.

5.2.2 Matching Network Architectures

In order to match the RGB (or MSI) scattering coefficients to the corresponding HSI coefficients, we mainly considered two different shallow CNN architectures. We refer to the first (and our original) network architecture as the No-Dilation-Together (NDT) matching network; its general architecture can be seen in Figure 5.4. It upsamples the combined number of input image and scattering channels to the combined number of the output channels via two convolutional blocks.

The network starts by reshaping the input data so that the image and scattering channels are in the same dimension; if the input scattering coefficients are of size $\left(\widehat{C}_1, S, \frac{M}{2^J}, \frac{N}{2^J}\right)$,it reshapes them to a tensor of size $\left(\widehat{C}_1 \cdot S, \frac{M}{2^J}, \frac{N}{2^J}\right)$. Then, it applies a convolutional layer (with bias) with window size 1×1 and Relu activation, followed by another 1×1 convolutional layer (with bias) and $\tanh$ activation. The output of this network is the predicted corresponding HSI scattering coefficients (which are also reshaped); thus, if the HSI has $\widehat{C}_2$ channels, then the output of the NDT matching network will be a tensor of size $\left(\widehat{C}_2 \cdot S, \frac{M}{2^J}, \frac{N}{2^J}\right)$. The number of convolutional filters applied in the first convolutional layer is a hyperparamater, but we choose it to be approximately twice as large as the number of convolutional filters in the second layer (which itself is determined by the number of channels of the output, i.e., is equal to $\widehat{C}_2 \cdot S$).

We call the second network architecture a Dilation-Separate (DS) matching network; its general architecture can be seen in Figure 5.5. Rather than matching all of the scattering coefficients of the input to those of the output, it instead matches the 0th layer coefficients of the input to the 0th layer coefficients of the output, and similarly for the 1st and 2nd layers. Additionally, the section of the network that matches the 0th layer coefficients uses a *dilated convolutional layer*. For an image $F : \{1, \ldots, M\} \times \{1, \ldots, N\} \to \mathbb{R}$, an $r \times r$ convolutional filter w (with $r \in \mathbb{N}$, odd), and $k \in \mathbb{N}$, a k-dilated convolution (actually k-dilated cross correlation) is defined by

$$F \star_k w(l) = \sum_i F(l + ki)w(i). \tag{5.1}$$

That is, each entry of the convolutional filter is applied to every k-th entry of F for each l. Dilated convolutions allow for a network to have a larger receptive field, i.e., incorporate information from farther away from the center of the filter, without an increase in needed computational resources [125].

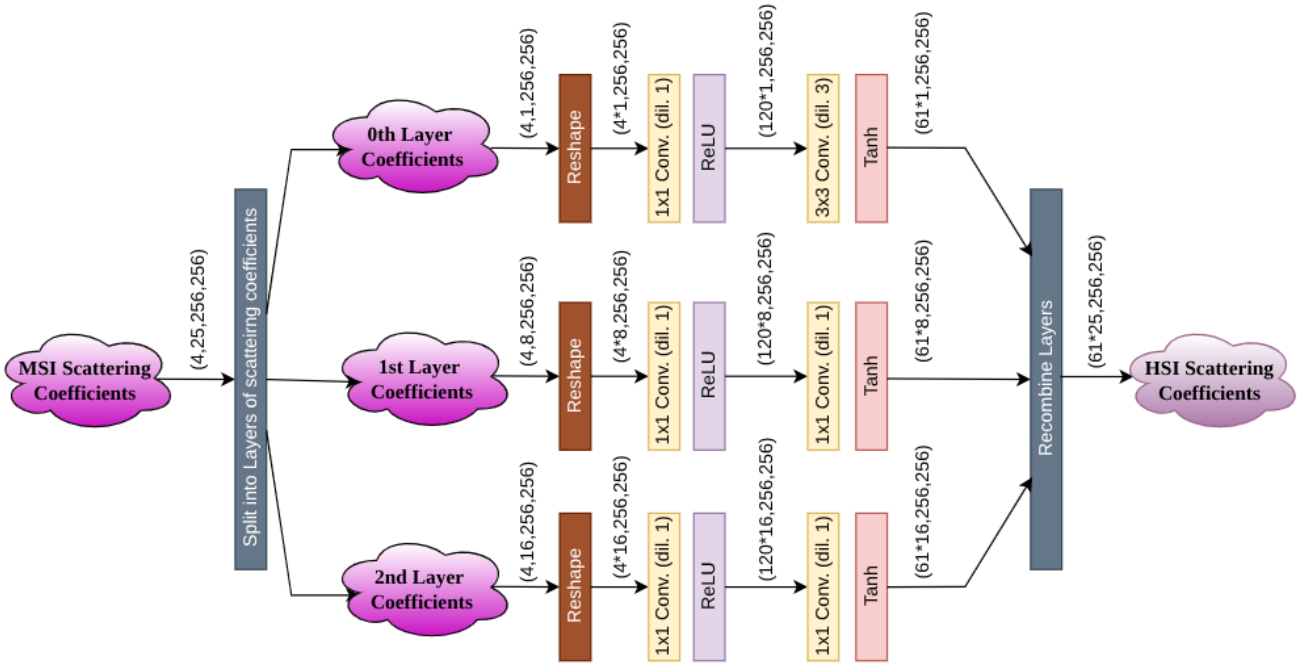


Figure 5.5: A diagram of the DS (Dilation-Separate) matching network architecture. The input is the scattering coefficients of an MSI, and are of size $(4, 25, 256, 256)$. The network matches between the 0th, 1st, and 2nd layers of scattering coefficients. It uses similar convolutional layers and activation functions as in the NDT matching networks, except for the 0th layer coefficients, for which a dilated convolutional layer is also used. At the end, the matched layer coefficients are recombined to get the predicted HSI scattering coefficients.

A DS matching network starts by separating the 0th, 1st, and 2nd layer scattering coefficients. It then reshapes these layers so that the image and scattering channels are in the same dimension, as in an NDT matching network. A 1×1 convolutional layer is then applied for each layer, followed by ReLU activation. For the 0th layer, a 3×3 convolutional layer with dilation 3 is then applied, while a 1×1 convolutional layer (with dilation 1) is applied to the 1st and 2nd layers. All convolutional layers for this network include a bias term. Finally, $\tanh$ activation is applied to each layer of coefficients, and the scattering coefficients are recombined to acquire the final output. As in the NDT architecture, the number of convolutional filters for each of the first applied convolutional layers is chosen to be approximately twice the number of the filters for each of the second convolutional layers.

Both network architectures are trained with Mean Square Error (MSE) loss. For the DS network, the loss is taken of the whole collection of scattering coefficients, rather than the 0th, 1st, and 2nd layers individually. In this way, the DS network is still trained to reconstruct the full scattering coefficients.

5.2.3 Inverse Network Architecture

As detailed in Section 2.4, an approximate inverse to the discretized WST embedding has been previously constructed and studied [14]. However, implementing the proposed inverse network in our setting leads to difficulties arising from computational restraints and the fact that our datasets contain relatively few images. In this section, we describe the above issues in more detail and propose a slight modification of the inverse network that is more amenable for hyperspectral reconstruction tasks.

5.2.3.1 Issues when Inverting the Discretized WST

While the scattering transform-based method constructed by Mallat and Angles for image generation [14] achieved good results, two aspects of the pipeline can lead to difficulties in using the proposed CNN architecture as an approximate inverse of the discretized WST. First, in order to transform the scattering transform coefficients into a (Gaussian) distribution, Mallat and Angle applied a whitening operator to the coefficients which also projected the data into a much smaller-dimensional space via PCA. However, while dimension reduction can make it easier to process the data, it also can lead to a significant loss of the information contained in the scattering coefficients. This is particularly an issue when dealing with datasets with large images but with only a few hundred samples, as the number of samples controls the maximum possible dimension that PCA can produce. For example, in the Hyper-Skin dataset in Section 2.5.3.2, there are 306 hyperspectral images of size $61 \times 1024 \times 1024$, and the scattering coefficients (with $J = 2$ dilations and $Q = 4$ rotations) would be of size $(61, 25, 256, 256)$. Since $306 \ll (61)(25)(256)(256)$, applying PCA would necessarily require discarding a large amount of information.

Second, the DCGAN used as the inversion network begins with a fully-connected layer to project the distribution back into the scattering space; see Figure 2.1 for a visualization of the network architecture. However, this layer can require significant computational resources in order to be trained. For example, with the Hyper-Skin dataset, each channel of the scattering coefficients is of size 256×256; thus, if the linear layer maps the lower-dimensional distribution of dimension α into a space of size $1024 \times 256 \times 256$ as suggested by Figure 2.1, the linear layer would require over $\alpha \cdot 67 \times 10^7$ parameters.

As the fully-connected layer in some sense exists to invert the PCA embedding, this leads to the question of whether these two components of the image generation pipeline are necessary when

attempting to invert the discretized WST. Inspired by the example "Regularized inverse of a scattering transform on MNIST" (see `https://www.kymat.io/gallery_2d/index.html`) provided by the creators of the discretized WST implementation Kymatio [13], we thus consider an inversion pipeline with no PCA embedding and no fully-connected layer.

5.2.3.2 Architecture of the Inverse Network

We use the implementation of the discretized WST as described in Section 2.4.1. This version calculates 2 layers of wavelet scattering transform coefficients using filters generated using J scales and Q rotations, where $J, Q \in \mathbb{N}$ are pre-chosen hyperparameters. The output scattering channels are then downsampled in each spatial dimension by a factor of 2^J. Thus, if the input image is of size $\widehat{C} \times M \times N$ where $\widehat{C}$ is the number of channels and M, N are the spatial dimensions, the WST output is of size $\left(\widehat{C}, S, \frac{M}{2^J}, \frac{N}{2^J}\right)$, where $S = 1 + JQ + \frac{J(J+1)Q^2}{2}$ is the number of scattering channels.

In order to invert the discretized WST, we first reshape the output so that the image and scattering channels are combined into one dimension. So, if the output is of size $\left(\widehat{C}, S, \frac{M}{2^J}, \frac{N}{2^J}\right)$, we reshape it to size $\left(\widehat{C} \cdot S, \frac{M}{2^J}, \frac{N}{2^J}\right)$. Then, as in the DCGAN, we apply J convolutional blocks, each consisting of a bilinear upsampling layer for each spatial dimension, a convolutional layer (without bias), batch normalization, and ReLU activation function. Then, a final convolutional block is applied, consisting of a convolutional layer (without bias), batch normalization, and $\tanh$ activation function. All of our convolutional layers use windows of size 3×3. A diagram of the inverse network architecture is shown in Figure 5.6, where the scattering transform was calculated using $J = 2$ dilations and $Q = 4$ rotations (in which case $S = 25$).

In the setting of hyperspectral reconstruction, our inverse network takes in the reshaped HSI

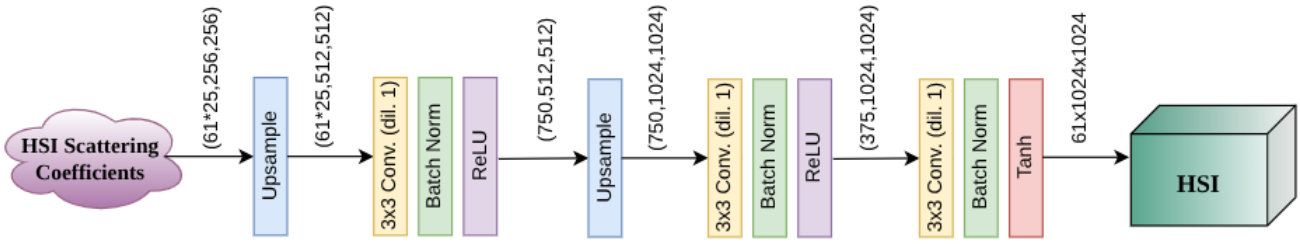


Figure 5.6: An example network architecture which inverts the discretized WST of an HSI. J convolutional blocks (consisting of spatial upsampling, a 3×3 convolutional layer with dilation 1 (and no bias), batch normalization, and ReLU activation) are applied, followed by a final convolutional block (consisting of a 3×3 convolutional layer with dilation 1 (and no bias), batch normalization, and $\tanh$ activation). In this example, a wavelet scattering transform with $J = 2$ dilations and $Q = 4$ dilations was used.

scattering coefficients and outputs the corresponding HSI. The inverse network is trained only on the actual HSI scattering coefficients, and hence its training is wholly independent from the training of the matching networks.

We considered two different loss functions when training this network. The first is l^1 loss, or Mean Absolute Error (MAE), which is defined in the usual way. For the second loss function, we first need to define a new metric: Let $H = \{h_{c,m,n}\}_{c,m,n} \in \mathbb{R}^{\hat{C} \times M \times N}$ be the original HSI and $\widetilde{H} = \{\widetilde{h}_{c,m,n}\} \in \mathbb{R}^{\hat{C} \times M \times N}$ be the predicted HSI. Also, let $h_{m,n} = \{h_{c,m,n}\}_c$ be the spectral vector of H at location (m, n) and similarly define $\widetilde{h}_{m,n}$. We can then define their cosine distance:

$$\text{cd}(h_{m,n}, \widetilde{h}_{m,n}) = 1 - \frac{h_{m,n} \cdot \widetilde{h}_{m,n}}{||h_{m,n}||_{l^2} ||\widetilde{h}_{m,n}||_{l^2}} \tag{5.2}$$

where $h_{m,n} \cdot \widetilde{h}_{m,n} \coloneqq \sum_{c=1}^{\hat{C}} h_{c,m,n} \widetilde{h}_{c,m,n}$ is the dot product and $\| \cdot \|_{l^2}$ is the standard l^2 norm. We then

define the cosine distance of $H, \widetilde{H}$ as the average:

$$\text{CD}(H, \widetilde{H}) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \text{cd}(h_{m,n}, \widetilde{h}_{m,n}). \tag{5.3}$$

Using CD as part of the loss function could result in gradient blow-ups due to the denominator in Equation (5.2). Thus, we define

$$\text{cd}_\delta(h_{m,n}, \widetilde{h}_{m,n}) := 1 - \frac{h_{m,n} \cdot \widetilde{h}_{m,n}}{(||h_{m,n}||_{l^2} + \delta)\left(||\widetilde{h}_{m,n}||_{l^2} + \delta\right)}. \tag{5.4}$$

for $\delta > 0$. We define CD_δ as the averages of the respective measurements. Then, our second loss function for the inverse network is $l^1 + \text{CD}_\delta$.

5.2.4 Splitting into Multiple Networks

Issues can arise when attempting to train the matching or inverse networks if there are not enough computational resources available. In order to overcome these barriers, we also explored training multiple pairs of matching and inverse networks, which each reconstruct different (but not necessarily disjoint) sections of the HSI spectra. Each of the matching networks, corresponding to a particular section, matches the RGB (or MSI) scattering coefficients to the corresponding HSI scattering coefficients from that section of the HSI. Then, the inverse network for the same section is applied to the predicted HSI scattering coefficients to recover the corresponding HSI section. The final result is then obtained by stacking the reconstructed channels from each section in the proper order.

Two examples of how to split an HSI into different sections of channels are shown in Figure 5.7. As shown, the HSI consists of two wavelength regimes: the visual (VIS) range (from 400 to 700 nm,

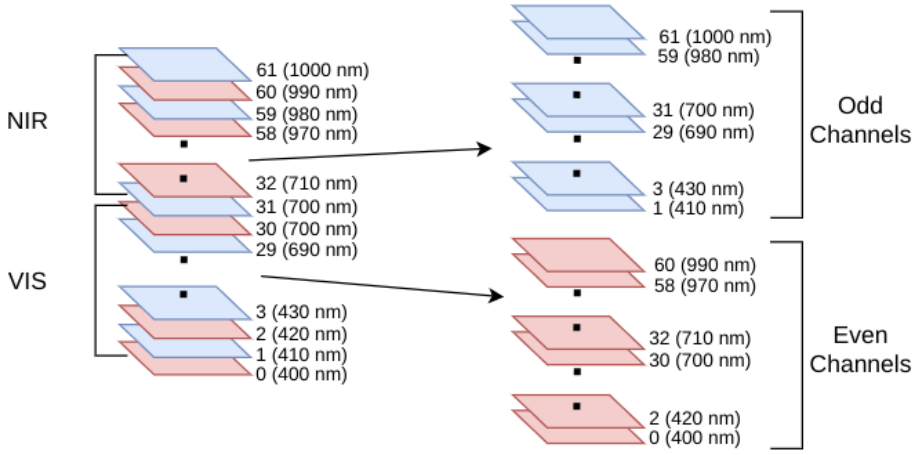


Figure 5.7: Examples of splitting HSI channels into different sections. On the left, the HSI are divided into the VIS (400–700 nm) and NIR (700–1000 nm) ranges. On the right, they are divided into even indexed and odd indexed channels. Note that, in both cases, the 700 nm channel was chosen to belong to both sections of the split.

in 10 nm increments), and the near-infrared (NIR) range (from 700 to 1000 nm, in 10 nm increments). Thus, a natural way of splitting our pipeline into two pairs of networks is to have one pair reconstruct the VIS range and the other reconstruct the NIR range. Alternatively, note that that we can give each channel of the HSI an index, and so another natural splitting divides the channels into those with even indices and those with odd indices. Note in both of these cases, however, that we choose to place the 700 nm index in both sections. We made this choice because having a common channel can be useful when comparing the reconstruction of the two sections. In this setting, when recombining the HSI sections after both have reconstructed, we choose to average the two predicted 700 nm channels.

Nevertheless, there are inherent trade-offs when choosing to split the Hyperscat pipeline into multiple pairs of networks. On the one hand, splitting does result in smaller networks that individually require less memory and are faster to train (see Table 5.36). However, some correlations between spectral channels in different sections are lost when splitting the networks, which can hamper performance by preventing the networks from learning said correlations. For example, in the VIS/NIR split, the

networks are unable to learn correlations between the two EM ranges, while in the even/odd split, the networks are unable to learn correlations between directly adjacent channels. Additionally, while the individual networks do require fewer computational resources when training, splitting does increase the number of networks that have to be trained. At the same time, when combined for inference, the split networks still require a total memory allocation similar to that of the non-split case.

5.2.5 Multi-Image Superresolution Network

As mentioned in Section 5.2.4, splitting the pipeline into multiple networks pairs can lead to correlations between the split HSI channels being lost during reconstruction. In an attempt to rectify this problem, we briefly explored applying an additional, multi-image superresolution (MISR) network as the final stage of reconstruction. Here, "MISR" refers to the idea of using information from multiple images to improve the resolution of a given image [91]. Similarly, we are attempting to perform spectral superresolution by merging groups of reconstructed channels into the full reconstructed HSI while improving the alignment between channels from different sections.

For a diagram of our MISR network architecture, see Figure 5.8. It is a two layer, fully connected network (with bias) with ReLU activation in the first layer and $\tanh$ activation in the second. Rather than applying the network to the whole image, however, it is only applied to each of the individual recombined spectra of the predicted HSI. The network is trained on the predicted spectra, and it is trained to produce spectra that more closely match the ground truth. MSE loss is used during training.

Additionally MISR networks can be trained to only improve the results for the spectra in a specified region of the HSI dataset (e.g., only on the face of an image in the Hyper-Skin dataset). This reduces training time, and it is useful in cases where some regions are either more important to

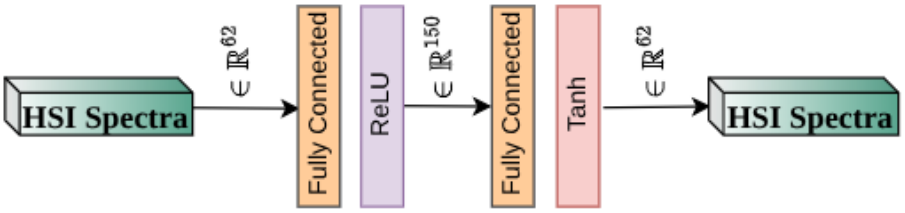


Figure 5.8: An example of a MISR network. It is a fully connected network used for improving alignment between channels in different sections. It takes in a predicted spectra created from two predicted sections and outputs a spectra closer to the ground truth.

reconstruct or have higher reconstruction error. Given a predicted HSI has shape $(\widehat{C}, M, N)$, there are at most $M \cdot N$ spectra in a single image on which the MISR network is trained.

5.3 Experimental Setup

In this section, we describe the setup for our numerical experiments of our hyperspectral reconstruction pipeline. First, we briefly discuss the datasets we on which we conducted our experiments, followed by a description of our baseline method and how we trained it. Next, we discuss some common implementation and training details of our own neural networks. Finally, we discuss several metrics we used for measuring our results.

5.3.1 Datasets

We conducted our hyperspectral reconstruction experiments on the Hyper-Skin and ARAD datasets (see Sections 2.5.3.2 and 2.5.3.1, respectively). For some of our experiments, we used our own training/validation/testing splits described in those sections. However, when we competed in the 2024 ICASSP Grand Challenge on Hyperspectral Skin Vision, the competition organizers tested our pipeline on their withheld testing set. Moreover, as they were only interested in our reconstruction

of the skin spectra of each participant, the challenge organizers also applied masks to the images when evaluating performance, and we did not have access to these masks during of after the competition. Thus, we distinguish between the *actual* Hyper-Skin dataset (with a training/testing split and with masks) with *our* Hyper-Skin dataset (with a training/validation/testing split). Additionally, for the ARAD dataset, we performed many of our experiments on the original training/validation split with no testing set. Thus, we also distinguish between the *actual* ARAD dataset (with the original training/validation split) and *our* ARAD dataset (with our training/validation/testing split) in a similar manner to Hyper-Skin.

5.3.2 Baseline Method

As a baseline method with which to compare our own model's performance, we used the MST++ network described in Section 2.5.3.3. As this method was already the top performing model for hyperspectral reconstruction at its time, we chose to only compare ourselves against this model. In our experiments, we aimed to train this network in a manner as close as possible to that used by the organizers of the ICASSP Grand Challenge [93]. So, on both datasets, we trained the model for 100 epochs with a batch size of 2 and l^1 loss function. Additionally, the Adam optimizer was used with an initial learning rate of 4×10^{-4} and gradient computation parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, along with a cosine annealing scheduler. We selected the same hyperparameters as used by the organizers, e.g., we chose $N_1 = N_2 = N_3 = 1$ and $N_S = 3$ (see Section 2.5.3.3 for more details).

However, there are several small changes we made while training the baseline method. First, during several experiments, we trained Hyper-Skin on *our* versions of Hyper-Skin and ARAD. In order to obtain a comparable network, we thus retrained MST++ using these validation and testing set

splits, and we chose the network that performed the best on the validation set during training as our comparison point. Additionally, due to computational constraints, MST++ was only trained on random 128×128 patches of the original Hyper-Skin dataset and during its original training on ARAD [29]. As we had similar constraints and in order to better compare our results against the Grand Challenge, we maintained this choice for the Hyper-Skin dataset. However, for ARAD, we chose to train the baseline on the full 512×512 channels as doing so was computationally feasible.

5.3.3 Implementation Details

For our experiments, we wrote all of our code in Python and implemented our neural networks using the package Pytorch. We trained the networks in parallel using at most 6 NVIDIA RTX A6000 GPUs, where the number of used GPUs depended on the batch size used during training. Additionally, while the original images take values in $[0, 1]$, we rescale them during training and inference so that they take values in $[-1, 1]$, since the output activation function of our networks, $\tanh$, takes values in $[-1, 1]$. After the inputs are fed through our pipeline, though, we shift the predicted and ground truth HSI back to the $[0, 1]$ range before we compute the loss or reconstruction metrics.

However, many implementation details, such as the optimizer used, number of epochs, batch size, initial learning rate, and use of weight decay vary according to the particular experiment. Thus, rather than listing them here, we shall note which choices have been made during each set of experiments.

5.3.4 Metrics

During our experiments, we considered two main metrics for measuring the performance of our proposed Hyperscat pipeline. The first metric is the Spectral Angle Mapper (SAM) score [46], which was also used during the aforementioned Grand Challenge [92,93]. This metric is used to compare the average errors between the predicted and the actual spectra of the HSI and thus captures how well the spectral (rather than spatial) information is reconstructed.

To define the measure, let $h_{m,n}$ and $\widetilde{h}_{m,n}$ be the real and predicted spectra, respectively, of some HSI. Then, the SAM score between these two spectra is

$$\text{sam}(h_{m,n}, \widetilde{h}_{m,n}) \coloneqq \cos^{-1}\left(\frac{h_{m,n} \cdot \widetilde{h}_{m,n}}{||h_{m,n}||_{l^2}||\widetilde{h}_{m,n}||_{l^2}}\right). \tag{5.5}$$

Thinking of the spectra as vectors in $\mathbb{R}^{\widehat{C}}$ (where $\widehat{C}$ is the number of channels of the HSI), the SAM score is just the angle between the two vectors. For H and $\widetilde{H}$, the real and predicted HSI, respectively, the SAM score is then given by the average of the SAM scores of the corresponding spectra:

$$\text{SAM}(H, \widetilde{H}) = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\text{sam}(h_{m,n}, \widetilde{h}_{m,n}). \tag{5.6}$$

Similarly to the case of cosine distance, we also define

$$\text{sam}_\delta(h_{m,n}, \widetilde{h}_{m,n}) \coloneqq \cos^{-1}\left(\frac{h_{m,n} \cdot \widetilde{h}_{m,n}}{(||h_{m,n}||_{l^2} + \delta)\left(||\widetilde{h}_{m,n}||_{l^2} + \delta\right)}\right), \tag{5.7}$$

and similarly $\text{SAM}_\delta(H, \widetilde{H})$, where $\delta > 0$ is some pre-chosen constant, when attempting to use SAM as part of a loss function.

If the spectra values are all non-negative, e.g., in $[0,1]$, then $\frac{h_{m,n}\cdot\widetilde{h}_{m,n}}{||h_{m,n}||_{l^2}||\widetilde{h}_{m,n}||_{l^2}}$ is non-negative and in the range $[0,1]$, with 0 meaning the vectors are orthogonal and 1 meaning that the vectors are parallel. Hence, the SAM score will be in the range $[0,\frac{\pi}{2}]$, where a lower score implies better performance.

The second metric we used is the Structural Similarity Index Measure (SSIM) [119]. This measure is commonly used when comparing images, and it is designed to capture how the human eye distinguishes between different pictures. SSIM measures how close the predicted and actual channels are, and thus captures how well the spatial (rather than spectral) information is reconstructed.

Given actual channel $H_c = \{h_{c,m,n}\}_{m,n}$ and predicted channel $\widetilde{H}_c = \{\widetilde{h}_{c,m,n}\}_{m,n}$, we define their SSIM score by

$$\text{ssim}(H_c, \widetilde{H}_c) = \frac{(2\mu_{H_c}\mu_{\widetilde{H}_c} + 0.01^2)(2\sigma_{H_c\widetilde{H}_c} + 0.03^2)}{(\mu_{H_c}^2 + \mu_{\widetilde{H}_c}^2 + 0.01^2)(\sigma_{H_c}^2 + \sigma_{\widetilde{H}_c}^2 + 0.03^2)} \tag{5.8}$$

where $\mu_{H_c}, \mu_{\widetilde{H}_c}$ are the samples means and $\sigma_{H_c}, \sigma_{\widetilde{H}_c}$ are the sample deviations of the channels. The SSIM of the real HSI $H = \{H_c : c = 1,\ldots,\widehat{C}\}$ and predicted HSI $\widetilde{H} = \{\widetilde{H}_c : c = 1,\ldots,\widehat{C}\}$ is then the average over all channels:

$$\text{SSIM}(H, \widetilde{H}) = \frac{1}{\widehat{C}}\sum_{c=1}^{\widehat{C}} \text{ssim}(H_c, \widetilde{H}_c). \tag{5.9}$$

Note that the SSIM of two images is always in the range $[-1,1]$, with a higher value indicating that the images are spatially closer. When reporting the SAM and SSIM results, we shall always report the average value of that metric on the specific data set, plus or minus the standard deviation.

Beyond reconstruction, we considered several different metrics for comparing the computational efficiency of our models against the baseline. During training, we calculated each network's Video

Random Access Memory (VRAM) usage, which captures how much memory is being used by the GPUs for training. Additionally, we used the summary function from the repository Torchinfo (see `https://github.com/TylerYep/torchinfo`) to calculate each network's multiply - accumulate (MAC) operations and size. The former measures how many times the networks perform operations like $ax + b$, i.e., performing multiplication and then addition, which are the vast majority of operations performed by a network. The size of a network encapsulates how much memory is needed to store the input, output, and intermediary data from each layer of calculations during inference.

5.4 Grand Challenge Experiments and Results

Our initial experiments for the hyperspectral reconstruction problem took place as part of the 2024 ICASSP Grand Challenge on Hyperspectral Skin Vision. As previously mentioned (see Sections 2.5.3.2 and 5.3.1), the goal of this competition was to reconstruct HSI from the (actual) Hyper-Skin dataset. As the competition organizers were chiefly concerned with the reconstruction of the skin spectra, they applied a mask when evaluating each competitor's model on the withheld testing set.

5.4.1 Our Models

At the time of the competition, we did not have the necessary computational resources to train the inverse network to reconstruct all 61 channels of the HSI. Thus, we explored the VIS/NIR splits and even/odd splits, as detailed in Section 5.2.4, and ultimately chose to split our networks to match even and odd channels due to the latter's observed improved performance. So, our main model consisted of applying the discretized WST to the input MSI, applying two NDT matching networks which separately reconstruct the even and odd HSI scattering channels, and then applying the two inverse net-

Model	Average SAM Score
MST++ (baseline)	0.1182 ± 0.0200
Model 1 (no MISR)	0.1201 ± 0.0116
Model 2 (MISR, 30 Epochs)	0.1183 ± 0.0124
Model 3 (MISR, 60 Epochs)	$\mathbf{0.1179 \pm 0.0129}$

Table 5.1: Results from the Grand Challenge on the masked, withheld Hyper-Skin testing set.

works to obtain the even and odd channels of the desired HSI. In some experiments, we also applied the MISR network in order to improve the alignment between adjacent channels.

For the competition, we trained on the full training set, i.e., we did not use validation. We trained the inverse networks for 150 epochs with l^1 loss function, and we trained the matching networks for 100 epochs with MSE loss function. We were allowed to submit three models for the competition; in our first, we only used the inverse and matching networks. For the second and third, at the end of the pipeline we applied the MISR network trained using MSE loss for 30 or 60 epochs, respectively. To attempt to focus on the skin spectra, we only trained the MISR network on spectra with at least 1 value in the the 70-th percentile of all values obtained by the 960 nm channel from the input MSI. The models were trained using the Adam optimizer with initial learning rate 10^{-3}, and with batch size 6.

5.4.2 Results

Our results for the Grand Challenge can be seen in Table 5.1. The competition organizers used MST++ as a baseline, and it was trained for 100 epochs on 128×128 patches of the full training set. The organizers tested all 4 of the presented models on the faces of the testing set by applying a withheld mask. The average SAM score (plus or minus the standard deviation) was then calculated for these masked images.

As seen in the table, our models performed similarly to the MST++ baseline, even though they were architecturally simpler. Additionally, while the MISR network led to slight improvement in performance, with Model 3 actually beating the baseline, the improvements were not substantial. Thus, as training the MISR network was both time consuming and was trained on the images predicted by the timeline (hence making it dependent on the particular trained inverse and matching networks) in our future experiments we ceased using this network. Ultimately, our numerical results put us in 5th place for the competition.

5.5 Optimization Experiments

In this section, we detail numerous experiments we conducted in our attempts to improve the models that we used for the Grand Challenge. First, we studied how our initial model architectures and training regimes are affected by not splitting the networks or by changing the loss function for the inverse network. Then, we studied how our models generalized to the ARAD dataset. After this, in order to determine what part of our method was leading to the biggest loss in performance, we studied how well different parts of our pipeline were learning the identity map. We then explored many different configurations of the matching networks, ultimately leading us to the DS architecture.

Next, we studied how changes in the training regime, e.g. training using validation, decreasing the initial learning rate, adding weight decay, varying the batch size, etc., affected performance. After this, we studied the combinations of our two inverse loss functions and two matching architectures on both datasets to see how they were improved using these training ideas. We briefly considered two alternate means of training: training the inverse and matching networks as one combined unit, and training the inverse network by computing the loss function in the scattering space. Finally, we studied

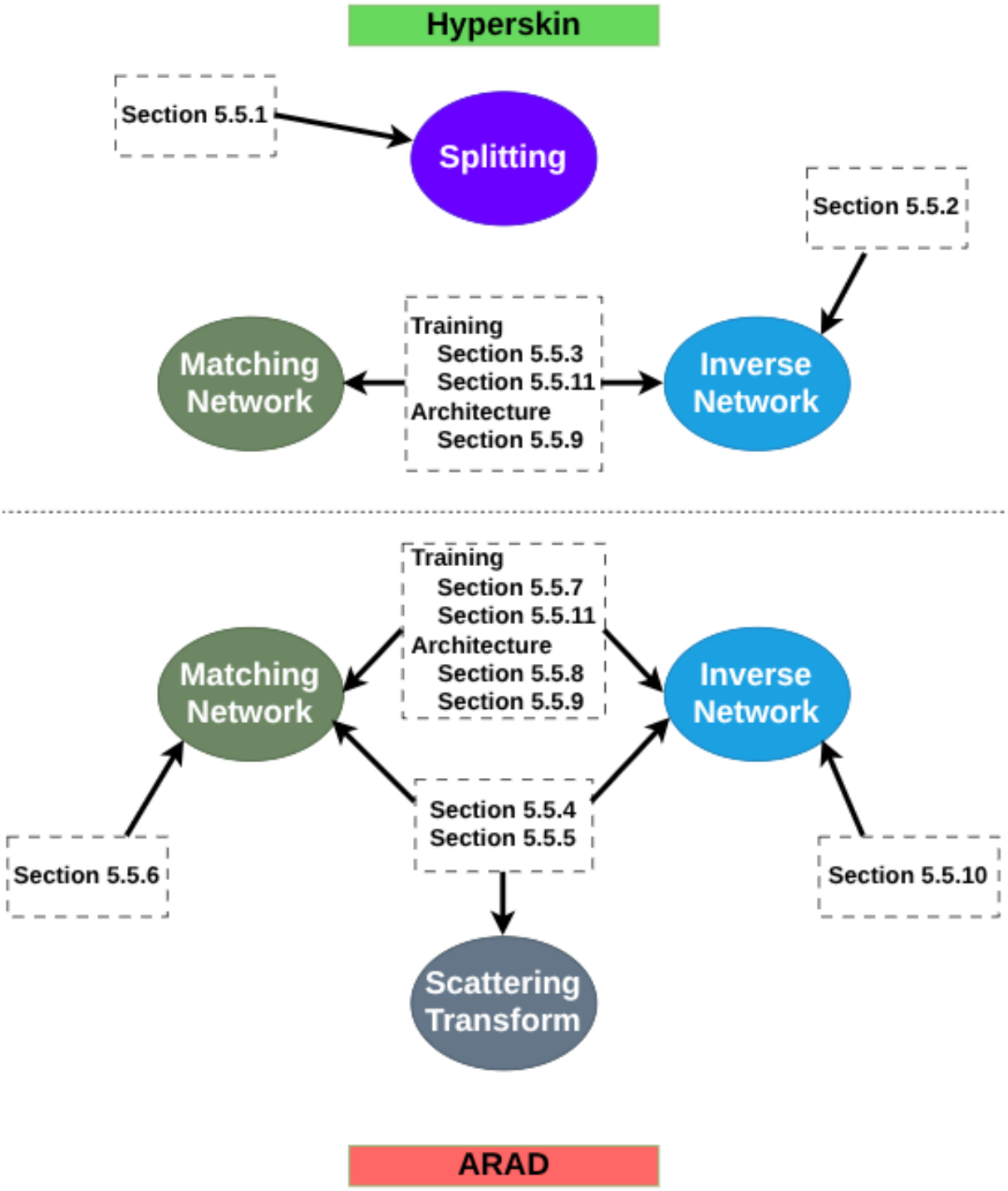


Figure 5.9: A diagram of our optimization experiments. Arrows point from boxes of subsections to the components(s) of the pipeline that were studied in those subsections. In the boxes, "Training" means that training parameters were optimized, while "Architecture" means that the network architectures were optimized. Hyper-Skin was used in the subsections above the dotted line, while ARAD was used in the subsections below the dotted line.

Model	Average SAM	Average SSIM
VIS/NIR Split	0.1397 ± 0.0092	0.9076 ± 0.0229
Even/Odd Split	0.1363 ± 0.0062	0.9104 ± 0.0227
No Split	0.1284 ± 0.0080	0.9142 ± 0.0218

Table 5.2: Results from our initial splitting experiments on our testing set for Hyper-Skin.

training the networks using half as many epochs and with different initial learning rates in order to create models that are fairer comparisons against the baseline. Unless otherwise specified, the results in this section are from our testing set for Hyper-Skin (with the training/validation/testing split) or the validation set for ARAD with the original training/validation split (i.e., no removed testing set). For the reader's convenience, Figure 5.9 provides a diagram of which components of the pipeline were studied in the following subsections of experiments.

5.5.1 Initial Splitting Experiments

Our first set of experiments concerned how performance of our model is affected by different splitting regimes, versus no splitting of networks. We trained and analyzed three different pipelines. The first split the matching and inverse networks to match the VIS and NIR channels separately, while the second split into even and odd channels. The third pipeline did not split the networks and hence reconstructed all channels at once. None of the pipelines used the MISR network. We trained these pipelines on our Hyper-Skin dataset, i.e., the version with the training/validation/testing split. All other training details, e.g. batch size, number of epochs, etc., remained the same as in Section 5.4.

Our average SAM results of the pipelines on our Hyper-Skin testing set are in Table 5.2. As can be seen, splitting into even and odd channel networks led to improved performance over splitting into VIS and NIR networks, while using no splits further improved results. This indicated that, in the

Loss Function	Average SAM	Average SSIM
l^1	0.1284 ± 0.0080	0.9142 ± 0.0218
$l^1 + \text{SAM}_{0.001}$	0.1350 ± 0.0078	0.8689 ± 0.0127
$l^1 + \text{CD}_{0.001}$	0.1332 ± 0.0094	0.8930 ± 0.0198

Table 5.3: Results from our initial inverse network loss function experiments on our testing set for Hyper-Skin.

Grand Challenge, we correctly chose to split into even and odd channels, and our results would have been furthered improved if we had been able to train a non-split pipeline. Note, however, that the results in this table are all worse than the results in Table 5.1. This can be explained by the fact that the testing sets used in the two tables are different, and that the the results in the first table were calculated by masking the non-skin spectra.

5.5.2 Initial Inverse Network Loss Function Experiments

Our next set of experiments concerned attempts to improve the SAM results of the reconstruction by modifying the loss function used for the inverse network. Originally, our inverse networks were trained using l^1 loss. However, this loss does not take into account the spectral information of the HSI, and real and predicted HSI can have small l^1 loss yet spectra which are dissimilar when considered as vectors.

Thus, we considered two more training regimes. In the first, the inverse was trained using $l^1 + \text{SAM}_{0.001}$ loss, where the second metric measures how close the spectra are. We also experimented with using $l^1 + \text{CD}_{0.001}$ loss, as cosine distance is highly correlated with SAM but comes with the added benefit that training with it is more stable, as SAM involves the concave function $\cos^{-1}$. We trained the non-split networks with these losses on our Hyper-Skin set, and we maintained all other training hyperparameters as in the previous experiments.

Model	Average SAM	Average SSIM
Hyperscat	0.1395 ± 0.0078	0.8412 ± 0.0401
MST ++	0.1400 ± 0.0058	0.9493 ± 0.0039

Table 5.4: Results from reconstructing the visual range HSI from Hyper-Skin RGB images. Results are from our testing set of Hyper-Skin.

The results from these experiments are in Table 5.3. As shown, both of the two new loss functions led to slightly worse results than training with l^1 loss. However, we opted to continue testing the $l^1 + \text{CD}_{0.001}$ loss in future experiments due to our desire to use the spectral information while training, and we will ultimately see that it can lead to improved results, even if it didn't in this instance. Additionally, we did not further consider $l^1 + \text{SAM}_{0.001}$ loss, both because it achieved worse results than using $CD_{0.001}$ and because it led to unstable training.

5.5.3 Generalization Experiments

Next, we explored how well our pipeline generalized to the ARAD dataset. First, rather than training directly on this new dataset, we extracted several images (99, 100, 386, 867, and 868) from ARAD of statues and murals that depicted people in a similar position as in the images in the Hyper-Skin dataset. Then, we fed them into our no-split, l^1 loss pipeline that was trained on Hyper-Skin to see how well it could reconstruct faces outside the particular setting in which the Hyper-Skin images were obtained.

However, the inputs of the ARAD dataset have 3 channels (as opposed to 4 in Hyper-Skin), while the outputs have 31 channels in the VIS range only (as opposed to 61 channels covering the VIS and NIR ranges). Thus, we retrained our pipeline to take in only the RGB channels of the Hyper-Skin dataset and output the visual range. We similarly retrained the MST++ network to do the same (with

Model	Image 99		Image 100		Image 386	
	SAM	SSIM	SAM	SSIM	SAM	SSIM
Hyperscat	0.5628	0.6605	0.7007	0.4958	0.6221	0.5773
MST ++	0.4027	0.9145	0.4992	0.7270	0.4572	0.8950

Model	Image 867		Image 868	
	SAM	SSIM	SAM	SSIM
Hyperscat	0.7449	0.3091	0.6921	0.4000
MST ++	0.5863	0.5765	0.5324	0.7836

Table 5.5: Generalization results obtained by inputting ARAD images into models (in Table 5.4) trained on Hyper-Skin.

the same training parameters as were used by the competition organizers). These experiments were conducted on our training/validation/testing splits of Hyper-Skin, and the average SAM and SSIM results (from the testing set) are in Table 5.4.

The generalization results (both SAM and SSIM) are presented in Table 5.5. For both Hyperscat and MST++, the SAM and SSIM results on the extracted ARAD images were poor, and they were significantly worse than the results on Hyper-Skin in Table 5.4. Moreover, in almost all cases, Hyperscat generalized worse than MST++. It should be reiterated, however, that neither model was trained on ARAD in these experiments. Thus, while this experiment cast doubt on models trained on Hyper-Skin being able to generalize to other, more complicated datasets, it did not preclude that these models could, if trained on ARAD, still perform well on the dataset.

5.5.4 Initial ARAD Experiments

Next, we experimented with training out Hyperscat pipeline directly on the ARAD dataset. In these initial experiments, we studied four different versions of our pipeline. For the first two, we used the same no-split networks as in the previous experiments, though now they match 3-channel RGB images to 31-channel HSIs. For the second model, we used the $l^1 + \text{CD}_{0.001}$ loss for the inverse

network.

Since the images in the ARAD dataset are noticeably smaller than those in Hyper-Skin, we were also able to explore the use of a three layer discretized WST, and we expected that the increase in extracted features would lead to improved performance. In this case, for an image F, number of dilations J, and number of rotations Q, the 3 layer scattering output $S_3[J,Q]F$ is the union of the following four sets:

$$\{F \star \phi_J\} \tag{5.10}$$

$$\{|F \star \psi_{j,q}| \star \phi_J\}_{\substack{1\leq j\leq J \\ 1\leq q\leq Q}} \tag{5.11}$$

$$\{||F \star \psi_{j,q}| \star \psi_{j',q'}| \star \phi_J\}_{\substack{1\leq j\leq j'\leq J \\ 1\leq q,q'\leq Q}} \tag{5.12}$$

$$\{|||F \star \psi_{j,q}| \star \psi_{j',q'}| \star \psi_{j'',q''}| \star \phi_J\}_{\substack{1\leq j\leq j'\leq j''\leq J \\ 1\leq q,q',q''\leq Q}} \tag{5.13}$$

Besides adding an extra layer of coefficients, we also replaced the requirement $j < j'$ with $j \leq j'$ (as otherwise, in the case $J = 2$, there would be no j'' bigger than j'). Choosing again the hyperparameters $J = 2$ and $Q = 4$, the output of an $\widehat{C} \times M \times N$ image is of size $\left(\widehat{C}, 121, \frac{M}{2^J}, \frac{N}{2^J}\right)$. In our third tested model, we used a 3 layer WST but with the same matching and inverse architecture as in model 1 (while increasing network layer sizes to accommodate the increased tensor sizes). Model 4 is the same as model 3 except it uses $l^1 + \text{CD}_{0.001}$ loss for the inverse network. We refer to these models as *Deeper Hyperscat*.

We maintained the same training regimen as in the previous Hyper-Skin experiments. That is, the inverse and matching networks were trained for 150 and 100 epochs, respectively, and the Adam optimizer was used with initial learning rate of 10^{-3}. We also trained MST++ for 100 epochs, but we chose to train it on the full 512×512 channels rather than on 128×128 patches as done in the Grand

Model	Average SAM	Average SSIM
Hyperscat (l^1)	0.1704 ± 0.0647	0.6822 ± 0.1621
Hyperscat ($l^1 + \text{CD}_{0.001}$)	0.1638 ± 0.0670	0.5945 ± 0.1507
Deeper Hyperscat (l^1)	0.1838 ± 0.0628	0.6442 ± 0.1565
Deeper Hyperscat ($l^1 + \text{CD}_{0.001}$)	0.1881 ± 0.0782	0.5485 ± 0.1549
MST ++	0.1390 ± 0.0776	0.8769 ± 0.1047

Table 5.6: Results from our initial ARAD experiments. Results are from the validation set.

Challenge. For all experiments, we used the original training/validation split for ARAD, and we tested our results on the validation set (note that the validation set was *not* used during training for validation and was instead kept separate).

The results of these experiments are presented in Table 5.6. There are several things to note. First, all of our models perform significantly worse than MST++ on ARAD, in contrast to HyperSkin. Second, the Deeper Hyperscat models perform worse than the normal Hyperscat models. Finally, in contrast to Table 5.3, the $l^1 + \text{CD}_{0.001}$ loss function does improve the reconstruction for SAM in this case, though with a concurrent decline in SSIM results. These results indicated that our pipeline needed improvement in order to achieve comparable results to MST++ on the more complicated ARAD dataset.

5.5.5 Identity Experiments

In an attempt to determine which portion of our pipeline to focus our efforts to improve Hyperscat, we conducted a variety of experiments testing how well different networks were learning the identity map. In the first set of experiments, we studied how close the output scattering coefficients of different models matched the actual HSI scattering coefficients. First, we looked at was how close the matching network outputs were to the ground truth HSI scattering coefficients. Next, we calculated

Dataset	Average Relative l^2		
	Matching	**Inverse**	**Hyperscat**
Hyper-Skin	0.0480 ± 0.0065	0.1176 ± 0.0193	0.1219 ± 0.0202
ARAD (2-layer scattering)	0.2399 ± 0.2433	0.2531 ± 0.2722	0.2808 ± 0.2686
ARAD (3-layer scattering)	0.2401 ± 0.2307	0.1614 ± 0.1287	0.2849 ± 0.2388

Table 5.7: Results from our experiments on how well different parts of our pipeline learn the identity for the discretized WST. Results are from our testing set of Hyper-Skin and the validation set of ARAD.

the scattering coefficients of the HSI which were predicted by the inverse network and measured how close they were to the actual HSI. Finally, we compared the scattering transform of the HSI predicted by the full Hyperscat model to the ground truth.

For these experiments, we tested the no-split networks used for the Hyper-Skin dataset on the testing set, as well as the 2-layer and 3-layer networks for the ARAD dataset (with l^1 loss for the inverse) on the validation set. In order to compare scattering coefficients, we used the relative l^2 error. Given ground truth data y and predicted data $\widetilde{y}$, the relative l^2 error is given by

$$\frac{\|y - \widetilde{y}\|_{l^2}}{\|\widetilde{y}\|_{l^2}}. \tag{5.14}$$

The results for these experiments are in Table 5.7. For Hyper-Skin, we see that the predicted scattering coefficients from the matching network are much closer to the ground truth coefficients than from the inverse and Hyperscat models. However, this ceases to be true for ARAD, and in fact all three models perform poorly in this case. This indicated to us that exploring improvement to the matching network would best assist with improving performance.

In order to verify this supposition, we next explored how well the inverse networks learned the identity mapping, i.e., how close the predicted HSI were to the actual HSI. For these experiments,

Layer / Loss	Dataset	Average SAM	Average SSIM	Average Relative l^2
2 / l^1	Hyper-Skin	0.1091 ± 0.0084	0.9150 ± 0.0259	0.1927 ± 0.0423
2 / l^1	ARAD	0.1077 ± 0.0476	0.7664 ± 0.1270	0.4995 ± 0.2535
2 / $l^1 + \text{CD}_{0.001}$	ARAD	0.0885 ± 0.0365	0.7571 ± 0.1085	0.4489 ± 0.1382
3 / l^1	ARAD	0.1186 ± 0.0479	0.8376 ± 0.0897	0.3499 ± 0.1098
3 / $l^1 + \text{CD}_{0.001}$	ARAD	0.1053 ± 0.0599	0.7675 ± 0.1162	0.4456 ± 0.1826

Table 5.8: Results from our experiments on how well different inverse models learn the identity mapping on HSI. Results are from our testing set of Hyper-Skin and the validation set of ARAD.

Layer / Loss	Dataset	Average SAM	Average SSIM	Average Relative l^2
2 / l^1	Hyper-Skin	0.1517 ± 0.0108	0.8117 ± 0.0540	0.3413 ± 0.0789
2 / l^1	ARAD	0.1523 ± 0.0610	0.7664 ± 0.1270	0.8617 ± 0.3583
2 / $l^1 + \text{CD}_{0.001}$	ARAD	0.1314 ± 0.0549	0.7571 ± 0.1085	0.8306 ± 0.2524
3 / l^1	ARAD	0.2062 ± 0.0717	0.4870 ± 0.1817	0.7037 ± 0.2480
3 / $l^1 + \text{CD}_{0.001}$	ARAD	0.1635 ± 0.1107	0.4843 ± 0.1847	0.8087 ± 0.2844

Table 5.9: Results from our experiments on how different inverse models learn the identity squared. Results are from our testing set of Hyper-Skin and the validation set of ARAD.

we tested both the non-split inverse network (with l^1 loss) on Hyper-Skin as well as the four inverse networks for the different Hyperscat models acting on ARAD that we considered in Table 5.6, i.e., training with l^1 or $l^1 + \text{CD}_{0.001}$ loss and training with 2 or 3 layer scattering coefficients. The results for these experiments are seen in Table 5.8. For the 2 and 3 layer inverses on ARAD, we also calculated the results from applying the identity squared, i.e., scattering transform followed by inverse followed by scattering transform followed by inverse; these results are in Table 5.9.

There are several things to note about these results. First, there is not a significant difference between the metrics for the inverse networks on Hyper-Skin versus ARAD. This supported our thoughts that attempting to improve the matching architecture on ARAD would be more fruitful, as there was more of a gap between the inverse and Hyperscat outputs on ARAD then there was on Hyper-Skin. Second, the inverse networks were worse for the 3-layer WST than for the 2-layer WST. Combining

Matching Model	Average SAM	Average SSIM
$(1,1)_1 \to (1,1)_2$	0.1704 ± 0.0647	0.6822 ± 0.1621
$(1,1)_1 \to (3,2)_2$	0.1759 ± 0.0661	0.6757 ± 0.1750
$(3,2)_1 \to (1,1)_2$	0.1988 ± 0.0663	0.6590 ± 0.1617
$(1,1)_1 \to (3,2)_2 \to (3,2)_3$	0.2210 ± 0.1139	0.6685 ± 0.1484
$(1,1)_1 \to (3,3)_2 \to (3,3)_3$	0.2172 ± 0.0912	0.6723 ± 0.1629
$(1,1)_1 \to (3,3)_2 \to (3,4)_3$	0.2215 ± 0.0826	0.6495 ± 0.1884
$(1,1)_1 \to (3,3)_2 \to (3,5)_3$	0.1912 ± 0.0758	0.6965 ± 0.1645

Table 5.10: Results from our experiments on different matching architectures using dilated convolutional layers. The metrics measure how good reconstruction is after combining these matching models with the previously trained inverse network (with l^1 loss), and the results are from the validation set of the actual ARAD dataset.

this observation with the full reconstruction results on ARAD, we see that the increase in extracted features was not leading to improved performance, and so we chose to stop considering 3-layer WSTs going forward.

5.5.6 Dilated and Separated Matching Architecture Experiments

In this section, we detail various experiments we conducted on different matching network architectures. In all of the experiments in this section, we have trained the matching network for 150 epochs with batch size 6 and an initial learning rate of 10^{-3}.

In our first set of experiments, we studied how the use of dilated convolutional layers would affect reconstruction. We tested various different configurations of what the dilation steps were, which layers used dilated convolutions, and how many layers were used in the matching network. We still used $\tanh$ as the activation function for the final layer, and all other layers used ReLU. For comparing our results, we fed the HSI scattering coefficients into the inverse network trained using l^1 loss to invert the 2-layer WST on ARAD and calculated the average SAM and SSIM on the actual validation set.

These results can be seen in Table 5.10. In this table and the following, $(a,b)_i$ refers to a con-

Model	Average SAM	Average SSIM
NDT	0.1704 ± 0.0647	0.6822 ± 0.1621
LSTT	0.1587 ± 0.0657	0.6711 ± 0.1725
LSTS	0.1599 ± 0.0645	0.6623 ± 0.1755
CSTT	0.1657 ± 0.0657	0.6711 ± 0.1762
LSTT ($l^1 + \text{CD}_{0.001}$ inverse loss)	0.1580 ± 0.0656	0.5637 ± 0.1584

Table 5.11: Results from experiments on matching networks that separately matched different layers or channels of scattering coefficients. Results are from the validation set of ARAD.

volutional layer in layer i which has a filter size a and a dilation factor b. We use arrows to indicate going from one layer to another. So, for example, the row $(1,1)_1 \rightarrow (3,2)_2$ means a 2 layer matching network whose first layer has window size 1 and dilation factor 1 (i.e., regular convolution) and the second layer has window size 3 and dilation factor 2. In this notation, $(1,1)_1 \rightarrow (1,1)_2$ is the standard (i.e., NDT) matching network. As seen in the table, none of these dilated matching networks beats the NDT matching network.

For our next set of experiments, we studied matching networks that match separate layers or channels of scattering coefficients, rather than matching all of the coefficients at once, where each of the separate matching components used the NDT architecture.. We expected such architectures would help the matching networks to more easily understand the relations within different scattering channels. For each separate part of the network, we still used ReLU as the first activation function and $\tanh$ as the final activation function.

In the first architecture, the matching network was trained to individually match 0th layer scattering coefficients to 0th layer coefficients, 1st layer to 1st layer, and 2nd layer to 2nd layer. However, after matching, the network would concatenate the separate layers into the full HSI scattering coefficients, and the network was trained on how well the full predicted coefficients matched the ground truth; we refer to this architecture as Layers Separated, Trained Together (LSTT). For the second ar-

Model	Average SAM	Average SSIM
$(1,1)_1 \to (3,2)_2$	0.1612 ± 0.0682	0.6626 ± 0.1747
$(1,1)_1 \to (3,3)_2$	0.1647 ± 0.0677	0.6581 ± 0.1672
$(1,1)_1 \to (3,4)_2$	0.1629 ± 0.0642	0.6585 ± 0.1688
$(1,1)_1 \to (3,5)_2$	0.1639 ± 0.0692	0.6535 ± 0.1653
$(1,1)_1 \to (3,6)_2$	0.1669 ± 0.0674	0.6423 ± 0.1652
$(1,1)_1 \to (3,2)_2 \to (3,2)_3$	0.1617 ± 0.7232	0.7051 ± 0.1717
$(1,1)_1 \to (3,2)_2 \to (3,3)_3$	0.1798 ± 0.0601	0.6658 ± 0.1596
$(1,1)_1 \to (3,2)_2 \to (3,4)_3$	0.1947 ± 0.0655	0.6474 ± 0.1540
$(1,1)_1 \to (3,2)_2 \to (3,5)_3$	0.1698 ± 0.0673	0.6898 ± 0.1622
$(1,1)_1 \to (3,2)_2 \to (3,6)_3$	0.1664 ± 0.0696	0.7084 ± 0.1633
$(1,1)_1 \to (3,3)_2 \to (3,2)_3$	0.1694 ± 0.0699	0.6453 ± 0.1545
$(1,1)_1 \to (3,3)_2 \to (3,3)_3$	0.1673 ± 0.0688	0.6567 ± 0.1537
$(1,1)_1 \to (3,3)_2 \to (3,4)_3$	0.1675 ± 0.0675	0.6506 ± 0.1518
$(1,1)_1 \to (3,3)_2 \to (3,5)_3$	0.1628 ± 0.0678	0.6675 ± 0.1443
$(1,1)_1 \to (3,3)_2 \to (3,6)_3$	0.1911 ± 0.0675	0.6462 ± 0.1444
$(1,1)_1 \to (3,4)_2 \to (3,2)_3$	0.1824 ± 0.0661	0.6484 ± 0.1523
$(1,1)_1 \to (3,4)_2 \to (3,3)_3$	0.1775 ± 0.0637	0.6533 ± 0.1539
$(1,1)_1 \to (3,4)_2 \to (3,4)_3$	0.1676 ± 0.0682	0.6581 ± 0.1511
$(1,1)_1 \to (3,4)_2 \to (3,5)_3$	0.1689 ± 0.0683	0.6591 ± 0.1514
$(1,1)_1 \to (3,4)_2 \to (3,6)_3$	0.1698 ± 0.0763	0.6573 ± 0.1595
$(1,1)_1 \to (3,5)_2 \to (3,2)_3$	0.1707 ± 0.0681	0.6364 ± 0.1499
$(1,1)_1 \to (3,5)_2 \to (3,3)_3$	0.1727 ± 0.0738	0.6397 ± 0.1600
$(1,1)_1 \to (3,5)_2 \to (3,4)_3$	0.1795 ± 0.0741	0.6302 ± 0.1593
$(1,1)_1 \to (3,5)_2 \to (3,5)_3$	0.1804 ± 0.0831	0.7077 ± 0.1562
$(1,1)_1 \to (3,5)_2 \to (3,6)_3$	0.1697 ± 0.0739	0.6441 ± 0.1467

Table 5.12: Results from experiments on 2 and 3 layer matching networks that combines the LSTT architecture with dilated convolutional layers. The results are measured by inserting the matching network results into the inverse network trained with l^1 loss. Results are from the validation set of ARAD.

chitecture, we trained 3 separate networks to separately match the 0th, 1st, and 2nd layers; we refer to this architecture as Layers Separated, Trained Separate (LSTS). Finally, for the third architecture, we matched the 25 scattering channels separately but concatenated at the end for training; this architecture is Channels Separated, Trained Together (CSTT).

The results of these experiments can be seen in Table 5.11. The LSTT model performed the best in terms of SAM and only slightly worse than the best in terms of SSIM. We also included the LSTT

Model	Average SAM	Average SSIM
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,2)_3 \rightarrow (3,2)_4$	0.1931 ± 0.0819	0.6858 ± 0.1567
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3 \rightarrow (3,3)_4$	0.1908 ± 0.0702	0.6400 ± 0.1415
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3 \rightarrow (3,4)_4$	0.1783 ± 0.0682	0.6539 ± 0.1408
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3 \rightarrow (3,5)_4$	0.1847 ± 0.0634	0.6482 ± 0.1406
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3 \rightarrow (3,6)_4$	0.1830 ± 0.0672	0.6167 ± 0.1427
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3 \rightarrow (3,7)_4$	0.1731 ± 0.0832	0.6902 ± 0.1408
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3 \rightarrow (3,4)_4$	0.1780 ± 0.0681	0.6395 ± 0.1433
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3 \rightarrow (3,5)_4$	0.1812 ± 0.0683	0.6464 ± 0.1369
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3 \rightarrow (3,6)_4$	0.2000 ± 0.0865	0.6672 ± 0.1482
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3 \rightarrow (3,7)_4$	0.1958 ± 0.0739	0.6806 ± 0.1530
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3 \rightarrow (3,8)_4$	0.1756 ± 0.0801	0.6546 ± 0.1426
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3 \rightarrow (3,3)_4$	0.1805 ± 0.0708	0.6378 ± 0.1397
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3 \rightarrow (3,4)_4$	0.1813 ± 0.0754	0.6893 ± 0.1473
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3 \rightarrow (3,5)_4$	0.1803 ± 0.0676	0.6490 ± 0.1343
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3 \rightarrow (3,6)_4$	0.1767 ± 0.0691	0.6474 ± 0.1429
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3 \rightarrow (3,7)_4$	0.1777 ± 0.0691	0.6444 ± 0.1360
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3 \rightarrow (3,4)_4$	0.1783 ± 0.0687	0.6360 ± 0.1427
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3 \rightarrow (3,5)_4$	0.1813 ± 0.0670	0.6421 ± 0.1370
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3 \rightarrow (3,6)_4$	0.1873 ± 0.0788	0.6864 ± 0.1449
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3 \rightarrow (3,7)_4$	0.1807 ± 0.0814	0.6495 ± 0.1528
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,6)_3 \rightarrow (3,9)_4$	0.1816 ± 0.0700	0.6298 ± 0.1522

Table 5.13: Results from experiments on 4 layer matching networks that combines the LSTT architecture with dilated convolutional layers. The results are measured by inserting the matching network results into the inverse network trained with l^1 loss. Results are from the validation set of ARAD.

results when the inverse trained with $l^1 + \text{CD}_{0.001}$ loss was used, which further improved the SAM results but worsened the SSIM score. Moreover, note that the LSTT model led to improved SAM and SSIM results over our normal, NDT matching architecture.

With this knowledge, we next explored combining the LSTT architecture with dilated convolutional layers. As in the experiments displayed in Table 5.10, we explored many different configurations of dilated convolutional layers, including the dilation factor and the number of total layers in the matching architecture. We studied 2, 3, and 4 layer matching networks, and we calculated our results using the inverse network trained with l^1 loss and the inverse trained with $l^1 + \text{CD}_{0.001}$ loss. The l^1

Model	Average SAM	Average SSIM
$(1,1)_1 \rightarrow (3,2)_2$	0.1610 ± 0.0670	0.5550 ± 0.1635
$(1,1)_1 \rightarrow (3,3)_2$	0.1556 ± 0.0650	0.5521 ± 0.1622
$(1,1)_1 \rightarrow (3,4)_2$	0.1532 ± 0.0630	0.5511 ± 0.1649
$(1,1)_1 \rightarrow (3,5)_2$	0.1559 ± 0.0658	0.5470 ± 0.1650
$(1,1)_1 \rightarrow (3,6)_2$	0.1552 ± 0.0640	0.5408 ± 0.1647
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,2)_3$	0.1760 ± 0.0784	0.6853 ± 0.1946
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,3)_3$	0.1721 ± 0.0647	0.5231 ± 0.1675
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,4)_3$	0.1747 ± 0.0641	0.5063 ± 0.1606
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,5)_3$	0.1736 ± 0.0760	0.6670 ± 0.1865
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,6)_3$	0.1733 ± 0.0793	0.6809 ± 0.1939
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,2)_3$	0.1675 ± 0.0726	0.5223 ± 0.1559
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3$	0.1696 ± 0.0704	0.5138 ± 0.1561
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3$	0.1662 ± 0.0708	0.5136 ± 0.1491
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,5)_3$	0.1626 ± 0.0718	0.5198 ± 0.1522
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,6)_3$	0.1737 ± 0.0675	0.5130 ± 0.1581
$(1,1)_1 \rightarrow (3,4)_2 \rightarrow (3,2)_3$	0.1707 ± 0.0660	0.5201 ± 0.1612
$(1,1)_1 \rightarrow (3,4)_2 \rightarrow (3,3)_3$	0.1666 ± 0.0632	0.5071 ± 0.1620
$(1,1)_1 \rightarrow (3,4)_2 \rightarrow (3,4)_3$	0.1758 ± 0.0718	0.5409 ± 0.1562
$(1,1)_1 \rightarrow (3,4)_2 \rightarrow (3,5)_3$	0.1661 ± 0.0717	0.5286 ± 0.1577
$(1,1)_1 \rightarrow (3,4)_2 \rightarrow (3,6)_3$	0.1699 ± 0.0772	0.5243 ± 0.1649
$(1,1)_1 \rightarrow (3,5)_2 \rightarrow (3,2)_3$	0.1640 ± 0.0691	0.4986 ± 0.1562
$(1,1)_1 \rightarrow (3,5)_2 \rightarrow (3,3)_3$	0.1671 ± 0.0723	0.5030 ± 0.1602
$(1,1)_1 \rightarrow (3,5)_2 \rightarrow (3,4)_3$	0.1655 ± 0.0657	0.5114 ± 0.1584
$(1,1)_1 \rightarrow (3,5)_2 \rightarrow (3,5)_3$	0.1778 ± 0.0869	0.6667 ± 0.1784
$(1,1)_1 \rightarrow (3,5)_2 \rightarrow (3,6)_3$	0.1662 ± 0.0757	0.5108 ± 0.1613

Table 5.14: Results from experiments on 2 and 3 layer matching networks that combines the LSTT architecture with dilated convolutional layers. The results are measured by inserting the matching network results into the inverse network trained with $l^1 + \text{CD}_{0.001}$ loss. Results are from the validation set of ARAD.

inverse loss results are in Table 5.12 (for 2 and 3 layer matching networks) and Table 5.13 (for 4 layer networks), and the $l^1 + \text{CD}_{0.001}$ inverse loss results are in Table 5.14 (for 2 and 3 layer networks) and Table 5.15 (for 4 layer networks).

There are many things to note about these results. Focusing first on the SAM results (which was our focus when running these experiments), recall that the average SAM for the standard LSTT network, which can be written as $(1,1)_1 \rightarrow (1,1)_2$, was 0.1587 when the inverse with l^1 loss was used

Model	Average SAM	Average SSIM
$(1,1)_1 \to (3,2)_2 \to (3,2)_3 \to (3,2)_4$	0.2162 ± 0.1066	0.6176 ± 0.1722
$(1,1)_1 \to (3,2)_2 \to (3,3)_3 \to (3,3)_4$	0.1877 ± 0.0744	0.5098 ± 0.1519
$(1,1)_1 \to (3,2)_2 \to (3,3)_3 \to (3,4)_4$	0.1736 ± 0.0716	0.5063 ± 0.1597
$(1,1)_1 \to (3,2)_2 \to (3,3)_3 \to (3,5)_4$	0.1849 ± 0.0718	0.5104 ± 0.1544
$(1,1)_1 \to (3,2)_2 \to (3,3)_3 \to (3,6)_4$	0.1713 ± 0.0681	0.4560 ± 0.1682
$(1,1)_1 \to (3,2)_2 \to (3,3)_3 \to (3,7)_4$	0.1861 ± 0.1042	0.6336 ± 0.1623
$(1,1)_1 \to (3,2)_2 \to (3,4)_3 \to (3,4)_4$	0.1766 ± 0.0737	0.4882 ± 0.1560
$(1,1)_1 \to (3,2)_2 \to (3,4)_3 \to (3,5)_4$	0.1791 ± 0.0719	0.4903 ± 0.1550
$(1,1)_1 \to (3,2)_2 \to (3,4)_3 \to (3,6)_4$	0.1897 ± 0.0864	0.6214 ± 0.1728
$(1,1)_1 \to (3,2)_2 \to (3,4)_3 \to (3,7)_4$	0.1812 ± 0.0701	0.6724 ± 0.1799
$(1,1)_1 \to (3,2)_2 \to (3,4)_3 \to (3,8)_4$	0.1784 ± 0.0843	0.5004 ± 0.1463
$(1,1)_1 \to (3,3)_2 \to (3,3)_3 \to (3,3)_4$	0.1808 ± 0.0750	0.4994 ± 0.1503
$(1,1)_1 \to (3,3)_2 \to (3,3)_3 \to (3,4)_4$	0.1973 ± 0.0866	0.6167 ± 0.1709
$(1,1)_1 \to (3,3)_2 \to (3,3)_3 \to (3,5)_4$	0.1726 ± 0.0651	0.4984 ± 0.1554
$(1,1)_1 \to (3,3)_2 \to (3,3)_3 \to (3,6)_4$	0.1755 ± 0.0730	0.5046 ± 0.1627
$(1,1)_1 \to (3,3)_2 \to (3,3)_3 \to (3,7)_4$	0.1749 ± 0.0762	0.4916 ± 0.1454
$(1,1)_1 \to (3,3)_2 \to (3,4)_3 \to (3,4)_4$	0.1776 ± 0.0733	0.4786 ± 0.1565
$(1,1)_1 \to (3,3)_2 \to (3,4)_3 \to (3,5)_4$	0.1803 ± 0.0783	0.4970 ± 0.1498
$(1,1)_1 \to (3,3)_2 \to (3,4)_3 \to (3,6)_4$	0.1842 ± 0.0833	0.6471 ± 0.1703
$(1,1)_1 \to (3,3)_2 \to (3,4)_3 \to (3,7)_4$	0.1851 ± 0.0902	0.4945 ± 0.1526
$(1,1)_1 \to (3,3)_2 \to (3,6)_3 \to (3,9)_4$	0.1813 ± 0.0739	0.4913 ± 0.1551

Table 5.15: Results from experiments on 4 layer matching networks that combines the LSTT architecture with dilated convolutional layers. The results are measured by inserting the matching network results into the inverse network trained with l^1+CD$_{0.001}$ loss. Results are from the validation set of ARAD.

and 0.1580 when the inverse with $l^1 + \text{CD}_{0.001}$ loss was used. Thus, by looking at Tables 5.12 and 5.13, we see that none of these dilated LSTT networks outperformed the standard LSTT network when the l^1 loss is used for the inverse. However, looking at Tables 5.14 and 5.15, we see several 2 layer dilated LSTT networks (where the inverse uses $l^1 + \text{CD}_{0.001}$ loss) that outperform both versions of the standard LSTT network. These results, for the first time in our experiments, indicated to us that the use of dilated convolutional layers could improve our results.

Next, when looking at the number of layers of the matching networks, note that, in most instances, increasing the number of layers lead to worse computational results regardless of which loss

Model	Average SAM	Average SSIM
$(1,1)_1 \to (3,2)_2$	0.1669 ± 0.0630	0.6673 ± 0.1710
$(1,1)_1 \to (3,3)_2$	0.1605 ± 0.0673	0.6611 ± 0.1749
$(1,1)_1 \to (3,4)_2$	0.1608 ± 0.0667	0.6645 ± 0.1704
$(1,1)_1 \to (3,5)_2$	0.1668 ± 0.0630	0.6669 ± 0.1720
$(1,1)_1 \to (3,6)_2$	0.1574 ± 0.0648	0.6627 ± 0.1680
$(1,1)_1 \to (3,2)_2 \to (3,2)_3$	0.1719 ± 0.0632	0.6699 ± 0.1579
$(1,1)_1 \to (3,3)_2 \to (3,2)_3$	0.1647 ± 0.0664	0.6586 ± 0.1618
$(1,1)_1 \to (3,3)_2 \to (3,3)_3$	0.1815 ± 0.0686	0.6505 ± 0.1504
$(1,1)_1 \to (3,3)_2 \to (3,4)_3$	0.1677 ± 0.0633	0.6415 ± 0.1500
$(1,1)_1 \to (3,3)_2 \to (3,5)_3$	0.1708 ± 0.0648	0.6618 ± 0.1546
$(1,1)_1 \to (3,3)_2 \to (3,6)_3$	0.1693 ± 0.0666	0.6573 ± 0.1556

Table 5.16: Results from experiments on 2 and 3 layer matching networks that combines the LSTT architecture with dilated convolutional layers for the 0th layer scatteirng coefficients only. The results are measured by inserting the matching network results into the inverse network trained with l^1 loss. Results are from the validation set of ARAD.

function was used for the inverse. Hence, we chose to stop considering 4 layer dilated matching networks in our future experiments. We similarly reduced the number of experiments we performed on 3 layer networks.

For our next set of experiments, we considered only using the dilated convolutional layers in the part of the LSTT network that matches the 0th layer coefficients. When looking at the discretized WST coefficients as displayed in Figure 5.3, we noticed that most of the coefficients in the first and second layers were small, and that the (relatively) large coefficients tended to lie on edges. Thus, we reasoned that using dilated convolutions for these layers could result in poor feature extraction by the network, as the large values would be skipped by the filters when applied to adjacent pixels. As this is not the case for the 0th layer (which is just a downsampled Gaussian blur of the original images), we experimented with only using dilated convolutions for this layer.

The results for these experiments when using the l^1 loss inverse network are in Table 5.16, while the results when using the $l^1 + \text{CD}_{0.001}$ loss are in Table 5.17. Comparing the first of these tables with

Model	Average SAM	Average SSIM
$(1,1)_1 \rightarrow (3,2)_2$	0.1529 ± 0.0633	0.5672 ± 0.1656
$(1,1)_1 \rightarrow (3,3)_2$	0.1507 ± 0.0658	0.5667 ± 0.1632
$(1,1)_1 \rightarrow (3,4)_2$	0.1537 ± 0.0646	0.5685 ± 0.1646
$(1,1)_1 \rightarrow (3,5)_2$	0.1543 ± 0.0623	0.5705 ± 0.1662
$(1,1)_1 \rightarrow (3,6)_2$	0.1530 ± 0.0633	0.5651 ± 0.1638
$(1,1)_1 \rightarrow (3,2)_2 \rightarrow (3,2)_3$	0.1681 ± 0.0666	0.5573 ± 0.1604
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,2)_3$	0.1652 ± 0.0656	0.5434 ± 0.1624
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,3)_3$	0.1826 ± 0.0701	0.5370 ± 0.1566
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,4)_3$	0.1587 ± 0.0654	0.5301 ± 0.1536
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,5)_3$	0.1607 ± 0.0677	0.5467 ± 0.1591
$(1,1)_1 \rightarrow (3,3)_2 \rightarrow (3,6)_3$	0.1610 ± 0.0683	0.5381 ± 0.1582

Table 5.17: Results from experiments on 2 and 3 layer matching networks that combines the LSTT architecture with dilated convolutional layers for the 0th layer scatteirng coefficients only. The results are measured by inserting the matching network results into the inverse network trained with $l^1 + \text{CD}_{0.001}$ loss. Results are from the validation set of ARAD.

Table 5.12, we see that the results are slightly improved when only using dilations for the 0th layer. However, when looking at the second of these tables and comparing it to Table 5.14, we get improved results; in particular, for the matching network $(1,1)_1 \rightarrow (3,3)_2$, we get the average SAM result of 0.1507, which is better than the SAM result of 0.1580 from the LSTT network trained without dilations. Thus, we chose these particular dilation parameters as our other matching architecture, which we previously referred to as the DS matching network (see Section 5.2.2).

5.5.7 Training Parameters Experiments

Now that we have improved results by using the DS matching architecture, we changed our focus to our training parameters. As a reminder, in all of the previous optimization experiments, we trained the inverse networks for 150 epochs and the matching networks for 100 epochs, both with a batch size of 6. They were trained using the Adam optimizer with initial learning rate 10^{-3}. Moreover, while

Inverse		DS Matching		Average SAM	Average SSIM
V	BS	V	BS		
✓	6		2	0.1776 ± 0.0952	0.7240 ± 0.1601
✓	6	✓	2	0.1743 ± 0.0909	0.7487 ± 0.1682
✓	2		2	0.1701 ± 0.0843	0.7288 ± 0.1634
✓	2	✓	2	0.1778 ± 0.0923	0.7574 ± 0.1744

Table 5.18: Results from our initial experiments on varying the batch size and introducing validation into our training regime. In the table, V means Validation (with a check mark indicating that the best performing model on the validation set was used during evaluation), and BS means batch size. Results are from the validation set of ARAD.

both our Hyper-Skin dataset and the ARAD datasets had validation sets, we were not using them for validation during training, i.e., saving the model that performed best on the validation set during the training process.

During these experiments, we aimed to determine whether changes in our training regime would lead to noticeable improvements in our numerical results on ARAD. In our first experiments, we introduced validation to our training procedure and we explored using a batch size of 2 for the matching networks in the pipeline. We trained the inverse network using l^1 loss (which we did for all other experiments in this section), and we trained DS matching networks. Our results (on the validation set) are in Table 5.18; we use V to mean validation and BS to mean batch size. A check mark means the best performing model during training on the validation set was used during evaluation, while no check mark means we used the model obtained after the last epoch.

Notice that all of these SAM results were significantly worse than our previous best result of 0.1507 (from Table 5.17). So, we also experimented with changing the number of epochs as this would help the networks to converge to a more optimal solution. We started these experiments by training the inverse network (with batch size 6) for 1000 epochs. These results are in Table 5.19. Here, we see much much better results, especially in the case where the final inverse network was used

Inverse		DS Matching		Average SAM	Average SSIM
V	BS	V	BS		
✓	6		2	0.1589 ± 0.0917	0.7346 ± 0.1618
✓	6	✓	2	0.1846 ± 0.1116	0.7639 ± 0.1718
	6		2	0.1337 ± 0.0739	0.7248 ± 0.1689
	6	✓	2	0.1433 ± 0.0729	0.7525 ± 0.1792

Table 5.19: Results from more experiments on varying the batch size and introducing validation into our training. In this case, we trained the inverse for **1000 epochs**. Results are from the validation set of ARAD.

Inverse			DS Matching			Average SAM	Average SSIM
V	BS	LR	V	BS	LR		
✓	6	10^{-4}		2	10^{-3}	0.1435 ± 0.0736	0.7021 ± 0.1593
✓	6	10^{-4}	✓	2	10^{-3}	0.1686 ± 0.0857	0.7274 ± 0.1694
	6	10^{-4}		2	10^{-3}	0.3407 ± 0.1858	0.6564 ± 0.1599
	6	10^{-4}	✓	2	10^{-3}	0.3887 ± 0.1887	0.6601 ± 0.1686

Table 5.20: Results from our experiments with setting the initial learning rate of the inverse network to 10^{-4}. The inverse network was trained for **1000 epochs**. Results are from the validation set of ARAD.

(i.e., the one obtained from the last epoch of training). Also, note that using the matching network that performs best on the validation set led to worse results in both case than using the final matching network; this is a pattern that will consistently appear throughout our experiments.

Next, we tested reducing the initial learning rate (LR) for the inverse network, as we suspected that our current learning rate of 10^{-3} was causing our networks to miss good local minima. Thus, we trained the inverse network for 1000 epochs with the Adam optimizer an initial learning rate of 10^{-4}; these results are in Table 5.20. Additionally, as we considered that a reduction in the initial learning rate could also require the network to be trained longer in order to find a good optimum, we also experimented with training the inverse for a further 1000 epochs, or 2000 total. These results are in Table 5.21.

We observed from these tables that training for 2000 epochs was resulting in good results, with

Inverse			DS Matching			Average SAM	Average SSIM
V	BS	LR	V	BS	LR		
✓	6	10^{-4}		2	10^{-3}	0.1344 ± 0.0708	0.7001 ± 0.1627
✓	6	10^{-4}	✓	2	10^{-3}	0.1383 ± 0.0694	0.7147 ± 0.1744
	6	10^{-4}		2	10^{-3}	0.1507 ± 0.0660	0.6292 ± 0.1457
	6	10^{-4}	✓	2	10^{-3}	0.1481 ± 0.0671	0.6274 ± 0.1599

Table 5.21: Results from our experiments with setting the initial learning rate of the inverse network to 10^{-4}. The inverse network was trained for **2000 epochs**. Results are from the validation set of ARAD.

Inverse			DS Matching			Average SAM	Average SSIM
V	BS	LR	V	BS	LR		
✓	6	10^{-4}		2	10^{-4}	0.1265 ± 0.0695	0.6868 ± 0.1779
✓	6	10^{-4}	✓	2	10^{-4}	0.1262 ± 0.0684	0.7017 ± 0.1685
	6	10^{-4}		2	10^{-4}	0.1498 ± 0.0660	0.6176 ± 0.1599
	6	10^{-4}	✓	2	10^{-4}	0.1456 ± 0.0658	0.6249 ± 0.1599

Table 5.22: Results from our experiments with setting the initial learning rate of the inverse and DS matching networks to 10^{-4}. The inverse and DS matching networks were trained for **2000 epochs**. Results are from the validation set of ARAD.

the result from using the best performing inverse network on validation and the final matching network being particularly good. As the (DS) matching network up to this point was still being trained with initial learning rate 10^{-3} for 100 epochs, we next retrained this network with initial learning rate 10^{-4} and for 2000 epochs. These results are in Table 5.22; here, we see that both cases in which the best performing inverse network on the validation set was used produced the best results up to this point in terms of SAM on ARAD.

However, we still sought further means of improvement. In classical optimization tasks, a regularizer is often added as part of the loss function in order to prevent the predicted values from growing too large and also to help prevent overfitting [56]. In modern machine learning, a similar notion is given by *weight decay* (WD), in which the learned weight parameters are forced to grow smaller over time to improve performance. This has been combined with the adaptive gradient descent algorithm Adam

Inverse				DS Matching				Average SAM	Average SSIM
V	**BS**	**LR**	**WD**	**V**	**BS**	**LR**	**WD**		
✓	6	10^{-4}	10^{-5}		2	10^{-4}	0	0.1320 ± 0.0716	0.7220 ± 0.1751
✓	6	10^{-4}	10^{-5}	✓	2	10^{-4}	0	0.1307 ± 0.0705	0.7371 ± 0.1649
	6	10^{-4}	10^{-5}		2	10^{-4}	0	0.5677 ± 0.2195	0.6690 ± 0.1644
	6	10^{-4}	10^{-5}	✓	2	10^{-4}	0	0.5987 ± 0.2196	0.6719 ± 0.1566
✓	6	10^{-4}	0		2	10^{-4}	10^{-5}	0.1254 ± 0.0678	0.7014 ± 0.1780
✓	6	10^{-4}	0	✓	2	10^{-4}	10^{-5}	0.1279 ± 0.0686	0.7074 ± 0.1723
	6	10^{-4}	0		2	10^{-4}	10^{-5}	0.1485 ± 0.0644	0.6315 ± 0.1637
	6	10^{-4}	0	✓	2	10^{-4}	10^{-5}	0.1495 ± 0.0652	0.6318 ± 0.1658
✓	6	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1246 ± 0.0678	0.7339 ± 0.1749
✓	6	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1308 ± 0.0705	0.7391 ± 0.1697
	6	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.5689 ± 0.2174	0.6780 ± 0.1627
	6	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.5794 ± 0.2169	0.6762 ± 0.1632

Table 5.23: Results from our experiments with training the inverse and DS matching networks with the AdamW optimizer and weight decay. The inverse and DS matching networks were trained for **2000 epochs**. Results are from the validation set of ARAD.

to create the AdamW optimizer, which combines the adaptive learning rate with weight decay [85].

Hence, we next experimented with incorporating weight decay in our training. We used the AdamW optimizer for training, with a weight decay factor of 10^{-5}; all other training parameters were kept the same as in the experiments in Table 5.22. Moreover, we experimented with training with weight decay with either the inverse network or the DS network, or both. The results of these experiments are in Table 5.23. We also retrained the inverse networks with a batch size of 2 and repeated the weight decay experiments (including the case where neither network is trained with weight decay); these results are in Table 5.24. Comparing these tables with Table 5.22, we see that weight decay marginally improved the overall SAM results, but only by about 0.0016 in the best improvement.

Inverse				DS Matching				Average SAM	Average SSIM
V	**BS**	**LR**	**WD**	**V**	**BS**	**LR**	**WD**		
✓	2	10^{-4}	0		2	10^{-4}	0	0.1426 ± 0.0785	0.7223 ± 0.1761
✓	2	10^{-4}	0	✓	2	10^{-4}	0	0.1402 ± 0.0757	0.7354 ± 0.1663
	2	10^{-4}	0		2	10^{-4}	0	0.2409 ± 0.1528	0.6866 ± 0.1730
	2	10^{-4}	0	✓	2	10^{-4}	0	0.2553 ± 0.1573	0.6895 ± 0.1623
✓	2	10^{-4}	10^{-5}		2	10^{-4}	0	0.1254 ± 0.0667	0.7035 ± 0.1735
✓	2	10^{-4}	10^{-5}	✓	2	10^{-4}	0	0.1271 ± 0.0650	0.7192 ± 0.1648
	2	10^{-4}	10^{-5}		2	10^{-4}	0	0.1458 ± 0.0809	0.7102 ± 0.1907
	2	10^{-4}	10^{-5}	✓	2	10^{-4}	0	0.1444 ± 0.0775	0.7194 ± 0.1861
✓	2	10^{-4}	0		2	10^{-4}	10^{-5}	0.1315 ± 0.0711	0.7356 ± 0.1771
✓	2	10^{-4}	0	✓	2	10^{-4}	10^{-5}	0.1400 ± 0.0759	0.7401 ± 0.1724
	2	10^{-4}	0		2	10^{-4}	10^{-5}	0.2424 ± 0.1469	0.7060 ± 0.1726
	2	10^{-4}	0	✓	2	10^{-4}	10^{-5}	0.2504 ± 0.1515	0.7067 ± 0.1677
✓	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1253 ± 0.0647	0.7168 ± 0.1758
✓	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1290 ± 0.0659	0.7228 ± 0.1709
	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1396 ± 0.0760	0.7290 ± 0.1912
	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1407 ± 0.0763	0.7313 ± 0.1853

Table 5.24: Results from more experiments with training the inverse and DS matching networks with the AdamW optimizer and weight decay. The inverse and DS matching networks were trained for **2000 epochs**, and both networks used batch size 2. Results are from the validation set of ARAD.

5.5.8 Training the Pipeline as One Network

We next briefly studied what would happen if, instead of training the inverse and DS matching networks as separate networks as we had done up to this point, we trained them as one combined network. This network would take in as input the RGB scattering coefficients and output the corresponding HSI (not the HSI scattering coefficients). While the separation of the networks not only allows us to tune the two networks separately but also treat them as objects that do separate tasks, we expected that combining them into one single network could result in improved outcomes as the "matching network" portion would learn the errors of the "inverse network" section and adjust its weights accordingly, and vice versa.

Model	Average SAM	Average SSIM
Best Validation	0.3114 ± 0.0860	0.2867 ± 0.0908
Final Epoch	0.3142 ± 0.0865	0.2741 ± 0.0819

Table 5.25: Results from experiments where the DS matching and inverse networks are combined and trained as one single network. Results are from the validation set of ARAD.

In these experiments, we trained the combined network for 2000 epochs on ARAD with loss function l^1, batch size 2, initial learning rate 10^{-4}, and weight decay 10^{-5}, in line with the previous section's experiments. We also considered both the network with the best performance on the validation set and the network obtained after the final epoch. The results are in Table 5.25. As can be seen, these experiments resulted in high SAM and low SSIM scores, and so we stopped studying this training regime.

5.5.9 Loss Function and Matching Architecture Comparisons

After our training parameter experiments, we next studied training the inverse network with the same training regime in Table 5.24 but with the $l^1 + \text{CD}_{0.001}$ loss function. That is, the inverse network was trained using the AdamW optimizer for 2000 epochs with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. We also used the DS matching network with the same training parameters as the inverse network. These results are in Table 5.26. Comparing with the results in Tables 5.23 and 5.24, we see a small improvement in the SAM results when using this loss function, as least when the inverse network which performed best on validation and the final DS matching network were used.

After these experiments, we retrained our original, NDT matching architecture using the same training regime that we used for the previous experiment. We tested combining this network with

Inverse				DS Matching				Average SAM	Average SSIM
V	BS	LR	WD	V	BS	LR	WD		
✓	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1232 ± 0.0686	0.7273 ± 0.1789
✓	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1244 ± 0.0687	0.7321 ± 0.1738
	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1240 ± 0.0669	0.6368 ± 0.1566
	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1280 ± 0.0679	0.6353 ± 0.1590

Table 5.26: Results from our experiments with training the inverse and DS matching networks with the AdamW optimizer and weight decay for 2000 epochs. The inverse network was trained with the $l^1 + \text{CD}_{0.001}$ loss function. Results are from the validation set of ARAD.

Inverse					NDT Matching				Average SAM	Average SSIM
Loss	V	BS	LR	WD	V	BS	LR	WD		
0 CD	✓	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1293 ± 0.0632	0.7324 ± 0.1587
0 CD	✓	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1328 ± 0.0675	0.7483 ± 0.1575
0 CD		2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1529 ± 0.0916	0.7307 ± 0.1644
0 CD		2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1776 ± 0.1141	0.7543 ± 0.1623
1 CD	✓	2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1165 ± 0.0641	0.7387 ± 0.1631
1 CD	✓	2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1219 ± 0.0684	0.7515 ± 0.1617
1 CD		2	10^{-4}	10^{-5}		2	10^{-4}	10^{-5}	0.1223 ± 0.0655	0.6508 ± 0.1432
1 CD		2	10^{-4}	10^{-5}	✓	2	10^{-4}	10^{-5}	0.1259 ± 0.0686	0.6676 ± 0.1460

Table 5.27: Results from our experiments with training the inverse and NDT matching networks with the same training regime used for the experiments in Table 5.26. Results are from the validation set of ARAD.

the two inverse networks trained with the two different loss functions. Based off of our previous experiments (see Tables 5.11, 5.16, and 5.17), we expected this to lead to worse reconstruction results.

Our results from these experiments on ARAD are in Table 5.27; we use 0 CD to mean the l^1 loss function was used, and we use 1 CD to mean the $l^1 + \text{CD}_{0.001}$ loss was used. While the l^1 loss function results for the NDT architecture were worse than the corresponding results for the DS architecture, the opposite was true when using the $l^1 + \text{CD}_{0.001}$ loss function. In fact, in the latter case, a new best reconstruction result for Hyperscat on ARAD was obtained when using the inverse network which performed best on the validation set and the final matching network.

Inverse		Matching		Average SAM	Average SSIM
Loss	**V**	**V**	**Arc.**		
0 CD	✓		DS	0.1199 ± 0.0084	0.9478 ± 0.0128
0 CD	✓	✓	DS	0.1285 ± 0.0106	0.9463 ± 0.0137
1 CD	✓		DS	0.1096 ± 0.0064	0.9614 ± 0.0091
1 CD	✓	✓	DS	0.1118 ± 0.0070	0.9619 ± 0.0088
0 CD	✓		NDT	0.1143 ± 0.0089	0.9517 ± 0.0123
0 CD	✓	✓	NDT	0.1213 ± 0.0093	0.9488 ± 0.0131
1 CD	✓		NDT	0.1165 ± 0.0641	0.7387 ± 0.1631
1 CD	✓	✓	NDT	0.1088 ± 0.0066	0.9625 ± 0.0083

Table 5.28: Results from our experiments with retraining the Hyperscat pipeline with the training regime used for ARAD on the **Hyper-Skin dataset**. We trained both kinds of networks for 900 epochs with batch size 2, initial learning rate 10^{-4}, and weigh decay factor 10^{-5}. Results are from our testing set of Hyper-Skin.

In order to check which matching network actually performs better, we repeated our experiments on Hyper-Skin. We considered the inverse network trained with the two different loss functions, and we considered both the NDT and DS matching architectures. We trained both kinds of networks with the AdamW optimizer for 900 epochs with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. As the final inverse network consistently led to worse hyperspectral reconstruction results for ARAD, we stopped calculating these results. Our results from these experiments on our testing set for Hyper-Skin are in Table 5.28 (here, Arc. stands for architecture). We continued to see that the NDT matching architecture led to better reconstruction than the DS architecture, and on Hyper-Skin, NDT did better for both inverse loss functions.

5.5.10 Scattering Loss Experiments

We next briefly explored changes to the DCGAN inverse network training procedure. When one normally trains a GAN, the generator is trained to match the dataset distribution, while the discriminator network is trained to distinguish between the original and generated data. However, in our

Inverse		Matching		Average SAM	Average SSIM
Loss	**V**	**V**	**Arc.**		
l^1	✓		DS	0.1293 ± 0.0723	0.7364 ± 0.1841
l^1	✓	✓	DS	0.1366 ± 0.0750	0.7406 ± 0.1788
MSE	✓		DS	0.1327 ± 0.0705	0.7326 ± 0.1837
MSE	✓	✓	DS	0.1381 ± 0.0723	0.7365 ± 0.1779
l^1	✓		NDT	0.1384 ± 0.0788	0.7450 ± 01687
l^1	✓	✓	NDT	0.1508 ± 0.0900	0.7593 ± 0.1653
MSE	✓		NDT	0.1363 ± 0.0702	0.7417 ± 0.1688
MSE	✓	✓	NDT	0.1386 ± 0.0718	0.7554 ± 0.1650

Table 5.29: Results from our experiments with retraining the inverse network on ARAD and training by calculating the loss of the scattering coefficients. We train both kinds of networks for 2000 epochs with batch size 2, initial learning rate 10^{-4}, and weigh decay factor 10^{-5}. Results are from the validation set of ARAD.

architecture, we only train a generator to generate the original data from the scattering coefficients, and no discriminator is trained. In these experiments, we trained the inverse network to directly match the *scattering coefficients* of the ground truth data, rather than just the original images. In this way, the scattering transform more clearly acts as the discriminator as it distinguishes between the scattering coefficients.

In order to train the inverse network using this method, we apply the discretized WST to the output of the inverse network during training, and we calculate the loss of the predicted scattering coefficients with the actual scattering coefficients. We conducted these experiments using both the l^1 (used for the normal inverse network) and MSE (used for training the matching network) loss functions for the scattering coefficients. We trained the inverse network for 2000 epochs with a batch size of 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5} on the ARAD dataset.

Our results from these experiments are in Table 5.29. Comparing with the results in Tables 5.24, 5.26, and 5.27, we see that training the inverse network with the loss calculated in the scattering space led to worse SAM and SSIM results in most cases than training in the image space. Thus, we no longer

Inverse			Matching		Average SAM	Average SSIM
Loss	**V**	**LR**	**V**	**Arc.**		
0 CD	✓	10^{-4}		DS	0.1276 ± 0.0673	0.7126 ± 0.1755
0 CD	✓	10^{-4}	✓	DS	0.1467 ± 0.0761	0.7263 ± 0.1712
0 CD	✓	10^{-3}		DS	0.1391 ± 0.0800	0.7385 ± 0.1783
0 CD	✓	10^{-3}	✓	DS	0.1480 ± 0.0815	0.7491 ± 0.1725
1 CD	✓	10^{-4}		DS	0.1214 ± 0.0663	0.7187 ± 0.1741
1 CD	✓	10^{-4}	✓	DS	0.1409 ± 0.0754	0.7369 ± 0.1692
1 CD	✓	10^{-3}		DS	0.1191 ± 0.0697	0.7260 ± 0.1776
1 CD	✓	10^{-3}	✓	DS	0.1425 ± 0.0791	0.7406 ± 0.1749
0 CD	✓	10^{-4}		NDT	0.1261 ± 0.0680	0.7207 ± 0.1760
0 CD	✓	10^{-4}	✓	NDT	0.1354 ± 0.0703	0.7380 ± 0.1650
0 CD	✓	10^{-3}		NDT	0.1681 ± 0.1203	0.7515 ± 0.1812
0 CD	✓	10^{-3}	✓	NDT	0.1622 ± 0.1082	0.7643 ± 0.1697
1 CD	✓	10^{-4}		NDT	0.1198 ± 0.0650	0.7290 ± 0.1759
1 CD	✓	10^{-4}	✓	NDT	0.1471 ± 0.0843	0.7459 ± 0.1659
1 CD	✓	10^{-3}		NDT	0.1186 ± 0.0672	0.7359 ± 0.1792
1 CD	✓	10^{-3}	✓	NDT	0.1200 ± 0.0687	0.7532 ± 0.1710

Table 5.30: Results from our experiments with training our models for half as long and by varying the initial learning rate on the ARAD dataset. We trained both kinds of networks for 1000 epochs with batch size 2 and weigh decay factor 10^{-5}. The matching networks were trained with initial learning rate 10^{-4}. Results are from our validation set of ARAD (where our testing set was withheld during training).

studied this training regime in our future experiments.

5.5.11 Training Time and Learning Rate Experiments

In our final group of optimization experiments, we returned to the training time parameter. The network architecture that we use as our baseline, MST++ (see Section 5.6 for comparisons of our pipeline results with this network), was trained for 100 epochs for the Hyper-Skin competition [93]. When retraining MST++ on our training/validation/testing splits of Hyper-Skin and ARAD, we similarly chose to train the network for 100 epochs. However, we have so far trained our own pipeline for 2000 epochs on ARAD and 900 epochs on Hyper-Skin.

Inverse			Matching		Average SAM	Average SSIM
Loss	**V**	**LR**	**V**	**Arc.**		
0 CD	✓	10^{-4}		DS	0.1165 ± 0.0079	0.9382 ± 0.0179
0 CD	✓	10^{-4}	✓	DS	0.1246 ± 0.0097	0.9373 ± 0.0182
0 CD	✓	10^{-3}		DS	0.1143 ± 0.0076	0.9678 ± 0.0057
0 CD	✓	10^{-3}	✓	DS	0.1214 ± 0.0087	0.9672 ± 0.0059
1 CD	✓	10^{-4}		DS	0.1160 ± 0.0065	0.9487 ± 0.0124
1 CD	✓	10^{-4}	✓	DS	0.1224 ± 0.0082	0.9490 ± 0.0124
1 CD	✓	10^{-3}		DS	0.1120 ± 0.0065	0.9706 ± 0.0039
1 CD	✓	10^{-3}	✓	DS	0.1193 ± 0.0078	0.9707 ± 0.0040
0 CD	✓	10^{-4}		NDT	0.1162 ± 0.0081	0.9447 ± 0.1522
0 CD	✓	10^{-4}	✓	NDT	0.1175 ± 0.0088	0.9498 ± 0.0136
0 CD	✓	10^{-3}		NDT	0.1123 ± 0.0075	0.9673 ± 0.0056
0 CD	✓	10^{-3}	✓	NDT	0.1140 ± 0.0082	0.9691 ± 0.0055
1 CD	✓	10^{-4}		NDT	0.1114 ± 0.0062	0.9515 ± 0.0112
1 CD	✓	10^{-4}	✓	NDT	0.1170 ± 0.0079	0.9560 ± 0.0099
1 CD	✓	10^{-3}		NDT	0.1076 ± 0.0064	0.9690 ± 0.0041
1 CD	✓	10^{-3}	✓	NDT	0.1127 ± 0.0073	0.9706 ± 0.0043

Table 5.31: Results from our experiments with training our models for half as long and by varying the initial learning rate on the Hyper-Skin dataset. We trained both kinds of networks for 450 epochs with batch size 2 and weigh decay factor 10^{-5}. The matching networks were trained with initial learning rate 10^{-4}. Results are from our testing set of Hyper-Skin.

In order to more fairly compare Hyperscat with MST++, we studied networks that are trained for roughly the same amount of time as the latter. For both datasets, MST++ takes roughly 24 hours to train for 100 epochs. Thus, in order to train for an equivalent amount of time, we train our pipeline for half as long as our previous experiments: 1000 epochs on ARAD and 450 epochs on Hyper-Skin. Since we were training for half as many epochs, we also experimented with whether an initial learning rate of 10^{-3} versus 10^{-4} for training the inverse network would lead to better reconstruction results. We maintained all of other training parameters, including training using the AdamW optimizer with batch size 2 and weight decay factor 10^{-5}. We also considered both the l^1 and $l^1 + \text{CD}_{0.001}$ loss functions for the inverse network as well as both the NDT and DS matching architectures.

Our results from the validation set of ARAD are in Table 5.30 (where in this case we trained the

model on our training/validation/testing split), and our results from our testing set of Hyper-Skin are in Table 5.31. Comparing to the results in Section 5.5.9, we see that, in most cases, training for half as many epochs led to only a small degradation in reconstruction results. Additionally, we see that using an initial learning rate of 10^{-3} versus 10^{-4} sometimes led to better results, but not consistently. Additionally, using 10^{-3} occasionally significantly worsened reconstruction. Due to this apparent instability with this learning rate, we use the results from training with an initial learning rate of 10^{-4} in our future comparisons in this chapter.

5.6 Comparisons to Baseline

In order to determine the relative performance of our models for hyperspectral reconstruction, we compare several of our best model architectures and training regimes from Section 5.5 to the MST++ baseline (see Section 5.3.2 for details on the training of this network). More specifically, we vary the loss function used for the inverse network (l^1 or $l^1 + \text{CD}_{0.001}$) as well as the matching network architecture (NDT or DS). Moreover, we compare out results on Hyper-Skin, both when our pipeline is trained for 900 epochs and for 450 epochs, as well as on ARAD, where our pipeline is trained for 2000 or 1000 epochs. All of our Hyperscat networks in this section are trained using the AdamW optimizer with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. We always use the best performing inverse network on the validation set and the final matching network obtained during training.

After these comparisons, we return to the setting of the ICASSP Grand Challenge (see Section 5.4) and Section 5.5.1 where we split the inverse and matching networks into multiple separate networks that reconstructed different parts of the HSI. We test both the even/odd split and the VIS/NIR

Inv. Loss	Mtc. Arc.	Ep.	Validation Set		Testing Set	
			Average SAM	Average SSIM	Average SAM	Average SSIM
0 CD	NDT	900	0.1185 ± 0.0063	0.9541 ± 0.0075	0.1143 ± 0.0089	0.9517 ± 0.0123
1 CD	NDT	900	0.1112 ± 0.0061	0.9586 ± 0.0073	0.1067 ± 0.0069	0.9614 ± 0.0092
0 CD	DS	900	0.1157 ± 0.0060	0.9529 ± 0.0066	0.1199 ± 0.0084	0.9478 ± 0.0128
1 CD	DS	900	0.1093 ± 0.0041	0.9622 ± 0.0050	0.1096 ± 0.0064	0.9614 ± 0.0091
0 CD	NDT	450	0.1183 ± 0.0053	0.9509 ± 0.0068	0.1162 ± 0.0081	0.9447 ± 0.1522
1 CD	NDT	450	0.1160 ± 0.0050	0.9523 ± 0.0070	0.1114 ± 0.0062	0.9515 ± 0.0112
0 CD	DS	450	0.1146 ± 0.0049	0.9439 ± 0.0086	0.1165 ± 0.0079	0.9382 ± 0.0179
1 CD	DS	450	0.1167 ± 0.0044	0.9508 ± 0.0063	0.1160 ± 0.0065	0.9487 ± 0.0124
MST++ (Baseline)			0.1062 ± 0.0047	0.9746 ± 0.0037	0.1114 ± 0.0058	0.9777 ± 0.0019

Table 5.32: Comparison of results from our Hyperscat pipeline and the MST++ baseline on the Hyper-Skin dataset. Both the inverse and matching networks were trained using the AdamW optimizer with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. The best performing inverse network and the final trained matching networks are used. MST++ was trained using the Adam optimizer for 100 epochs with batch size 2, initial learning rate 4×10^{-4}, and on 128×128 slices of the HSI.

split on Hyper-Skin, and we consider both matching architectures and training for either 900 or 450 epochs with the same training parameters as the non-split case. We end this section with a comparison of the computational efficiency metrics of our inverse and matching networks (both split and non-split) with the MST++ network.

5.6.1 Hyper-Skin Results

Our comparison results on the Hyper-Skin dataset are in Table 5.32 (we abbreviate inverse as Inv., matching as Mtc., and epochs as Ep.). Whereas we only measured the performance of our Hyperscat pipeline on the testing set during our optimization experiments (Section 5.5), here we present the performance on both the validation and testing sets. Comparing first on the validation set, we see that none of our pipelines outperformed MST++ in terms of average SAM or average SSIM, though using the DS matching architecture and $l^1 + \text{CD}_{0.001}$ loss for the inverse network resulted in our best

Figure 5.10: Example reconstructed HSI channels from our pipelines and MST++ on a test image from the Hyper-Skin dataset. Each column is a channel measured from the same wavelength. The Hyperscat reconstructions are denoted by (inverse loss function)/(matching architecture), and the Hyperscat networks were trained for 450 epochs.

performance. It should also be noted that training for 450 versus 900 epochs slightly worsened reconstruction, and that the DS architecture led to better performance than the NDT architecture.

On the testing set, however, many of these patterns are flipped. First, when training for 900 epochs and using the $l^1 + \text{CD}_{0.001}$ loss function, we observe better performance on SAM than the MST++ network, though still with slightly worse SSIM performance. Additionally, Hyperscat trained for 900 epochs led to much improved performance versus training for 450 epochs, and the DS architecture performed worse than the NDT architecture. This suggests both that training for longer improves generalization (as to be expected) and that the NDT matching architecture may lead to better generalization results.

A qualitative comparison of our reconstruction pipelines can be seen in Figure 5.10. The top row consists of 5 channels of a ground truth image from our Hyper-Skin testing set, while the next 4 rows show our reconstructions of the same 5 channels from our 4 Hyperscat pipelines that were trained for 450 epochs in Table 5.32. The bottom row shows the channel reconstructions from MST++. As can be seen, our reconstruction is qualitatively close to the ground truth, though Hyperscat's channels are slightly darker in general and have some noticeable white patches on the bottom.

5.6.2 ARAD Results

Our comparison results on the ARAD dataset are in Table 5.33. While we conducted most of our experiments on the original training/validation split of ARAD in Section 5.5 (except for the final training time experiments), in this section we trained the networks on our training/validation/testing split and report both the validation and testing set results. On the validation set, we see that all of our networks performed a small amount worse than MST++ in terms of SAM, but they were much

Inv. Loss	Mtc. Arc.	Ep.	Validation Set		Testing Set	
			Average SAM	Average SSIM	Average SAM	Average SSIM
0 CD	NDT	2000	0.1196 ± 0.0678	0.7324 ± 0.1726	0.1362 ± 0.0817	0.6537 ± 0.2199
1 CD	NDT	2000	0.1205 ± 0.0680	0.7060 ± 0.1563	0.1339 ± 0.0823	0.6419 ± 0.2077
0 CD	DS	2000	0.1209 ± 0.0694	0.7187 ± 0.1789	0.1377 ± 0.0853	0.6407 ± 0.2380
1 CD	DS	2000	0.1216 ± 0.0670	0.7163 ± 0.1733	0.1389 ± 0.0852	0.6296 ± 0.2232
0 CD	NDT	1000	0.1261 ± 0.0680	0.7207 ± 0.1760	0.1436 ± 0.0825	0.6353 ± 0.2197
1 CD	NDT	1000	0.1198 ± 0.0650	0.7290 ± 0.1759	0.1379 ± 0.0811	0.6422 ± 0.2181
0 CD	DS	1000	0.1276 ± 0.0673	0.7126 ± 0.1755	0.1442 ± 0.0857	0.6297 ± 0.2330
1 CD	DS	1000	0.1214 ± 0.0663	0.7187 ± 0.1741	0.1392 ± 0.0857	0.6356 ± 0.2275
MST++ (Baseline)			0.1121 ± 0.0741	0.9289 ± 0.0878	0.1307 ± 0.0906	0.8940 ± 0.1369

Table 5.33: Comparison of results from our Hyperscat pipeline and the MST++ baseline on the ARAD dataset. Both the inverse and matching networks were trained using the AdamW optimizer with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. The best performing inverse network and the final trained matching networks are used. MST++ was trained using the Adam optimizer for 100 epochs with batch size 2, initial learning rate 4×10^{-4}, and on the full 512×512 channels.

(≈ 0.15-0.2 points) worse on SSIM. Additionally, our pipelines trained for 1000 epochs performed slightly worse than the 2000 epoch pipelines.

Every presented network performs worse on the testing set than the validation set in terms of both SAM and SSIM. However, the reduction in performance from the validation set to the testing set is slightly less severe for our networks than the MST++ network, suggesting once again that our networks may generalize better. Finally, we once again see a larger gap on the testing set between our pipelines trained for 1000 epochs versus 2000 epochs.

A qualitative comparison of our reconstruction pipelines can be seen in Figure 5.11. As can be seen, the reconstructed images are more clearly visually distinct from the ground on ARAD than they were on Hyper-Skin. There are both spatial inaccuracies, such as the noisy sky in the top of the channels reconstructed using the NDT matching architectures, as well as spectral errors, such as each reconstructed 650 nm channel being generally lighter in color than the ground truth.

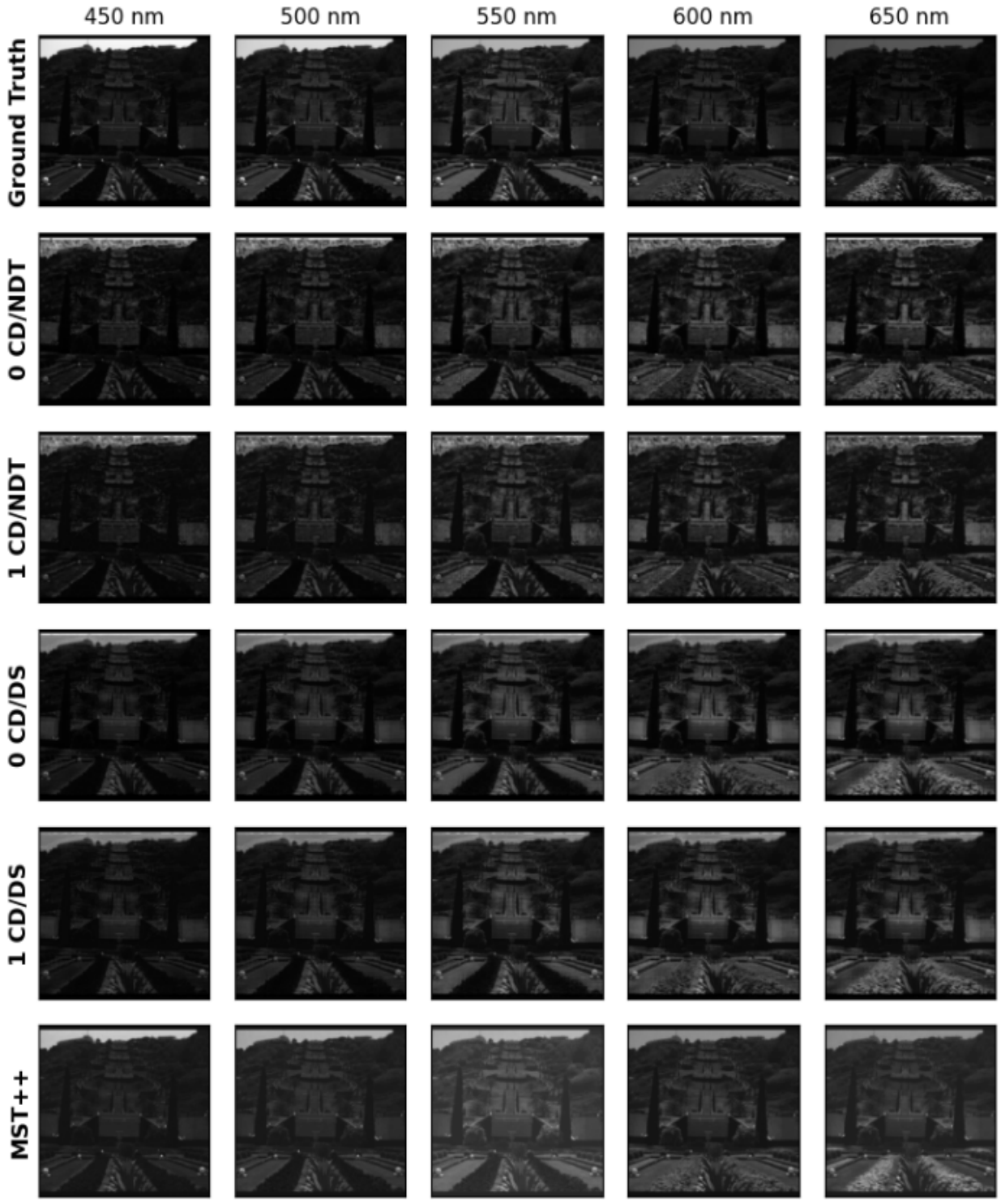


Figure 5.11: Example reconstructed HSI channels from our pipelines and MST++ on a test image from the ARAD dataset. Each column is a channel measured from the same wavelength. The Hyperscat reconstructions are denoted by (inverse loss function)/(matching architecture), and the Hyperscat networks were trained for 1000 epochs.

Mtc. Arc.	Ep.	Validation Set		Testing Set	
		Average SAM	Average SSIM	Average SAM	Average SSIM
NDT	900	0.1217 ± 0.0081	0.9596 ± 0.0066	0.1240 ± 0.0091	0.9592 ± 0.0096
DS	900	0.1175 ± 0.0047	0.9607 ± 0.0055	0.1222 ± 0.0077	0.9587 ± 0.0096
NDT	450	0.1317 ± 0.0095	0.9532 ± 0.0069	0.1280 ± 0.0094	0.9484 ± 0.0135
DS	450	0.1206 ± 0.0045	0.9493 ± 0.0073	0.1192 ± 0.0078	0.9445 ± 0.0147

Table 5.34: Comparison of results from splitting our Hyperscat pipeline to reconstruct the even and odd channels of the Hyper-Skin dataset separately. Both the inverse and matching networks were trained using the AdamW optimizer with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. The best performing inverse network and the final trained matching networks are used.

Mtc. Arc.	Ep.	Validation Set		Testing Set	
		Average SAM	Average SSIM	Average SAM	Average SSIM
NDT	900	0.1176 ± 0.0070	0.9622 ± 0.0055	0.1147 ± 0.0087	0.9664 ± 0.0064
DS	900	0.1204 ± 0.0049	0.9588 ± 0.0067	0.1173 ± 0.0085	0.9640 ± 0.0072
NDT	450	0.1140 ± 0.0076	0.9582 ± 0.0051	0.1224 ± 0.0091	0.9566 ± 0.0105
DS	450	0.1177 ± 0.0065	0.9463 ± 0.0079	0.1202 ± 0.0080	0.9516 ± 0.0108

Table 5.35: Comparison of results from splitting our Hyperscat pipeline to reconstruct the VIS and NIR channels of the Hyper-Skin dataset separately. Both the inverse and matching networks were trained using the AdamW optimizer with batch size 2, initial learning rate 10^{-4}, and weight decay factor 10^{-5}. The best performing inverse network and the final trained matching networks are used.

5.6.3 Splitting Results

Our results on the Hyper-Skin dataset for splitting the Hyperscat networks to reconstruct the even and odd channels separately are in Table 5.34, and our results when splitting them to reconstruct VIS and NIR channels are in Table 5.35. Comparing between these tables first, we see that, contrary to the ICASSP Grand Challenge (Table 5.1) and our initial splitting experiments (Table 5.2), splitting into VIS and NIR almost always produced better reconstruction results than splitting into even and odd. This is consistent across the number of epochs, matching architectures, and data set, suggesting that, if splitting is necessary, one should split the inverse and matching networks to reconstruct the VIS

and NIR channels if one is also training with our optimized training regimes.

Next, comparing these tables with Table 5.32, we see that choosing to split the networks into VIS and NIR leads to only a small reduction in performance when compared to the non-split networks (with the even and odd splits leading to more degradation). Indeed, some of the results when the NDT matching networks are used in the VIS/NIR split are as good or better than the corresponding results for the non-split networks. Though we did not experiment with training the split inverse networks with the $l^1 + \text{CD}_{0.001}$ loss functions, we expect that doing so would also lead to only a small performance loss when compared to the non-split networks.

5.6.4 Computational Efficiency Results

When calculating the computational efficiency metrics for our networks, we measured all metrics using a batch size of 1. We also measured the VRAM used during training using *half-precision* calculations (i.e., 16 bit), rather than *full-precision* calculations (i.e., 32 bit) like we used for all of our other training experiments, as this is how the calculations were performed in [116]. All other training parameters that we used in Section 5.6 are maintained, including using the AdamW optimizer with initial learning rate 10^{-4} and weight decay factor 10^{-5}.

Our computational efficiency results for our networks and MST++ on Hyper-Skin are in Table 5.36, and the results on ARAD are in Table 5.37. Note that, even when the metrics of the inverse and a matching network are combined, our Hyperscat pipeline has a significantly smaller size than MST++ on both datasets. However, the combined MACs are much larger, and so our pipeline presents a tradeoff during inference of smaller memory requirements but increased time required for calculations (note, though, that the total MAC operations when compared to MST++ for the DS matching

Network	Size (GBs)	MACs	VRAM (GBs)
Inverse	10.91	$5.57e12$	31.67
NDT Matching	2.42	$0.32e12$	8.24
DS Matching	2.41	$0.17e12$	7.88
MST++ (Baseline)	89.72	$1.17e12$	43.37
Inverse (Split)	5.59	$1.47e12$	17.04
NDT Matching (Split)	1.22	$0.09e12$	5.18
DS Matching (Split)	1.22	$0.05e12$	4.98

Table 5.36: Computational efficiency calculations for networks trained on the Hyper-Skin dataset (1024×1024 channels). Hyperscat has better results for size and VRAM, though is worse for MACs.

Network	Size (GBs)	MACs	VRAM (GBs)
Inverse	1.41	$366.32e9$	4.46
NDT Matching	0.31	$20.93e9$	1.56
DS Matching	0.31	$11.01e9$	1.50
MST++	11.40	$77.84e9$	5.92

Table 5.37: Computational efficiency calculations for networks trained on the ARAD dataset (512×512 images).

architecture is only slightly more than half that for the NDT matching architecture). Additionally, while the combined VRAM requirements of the inverse and a matching network are approximately equal to the VRAM needed for training MST++, the fact that these networks are trained separately means that less total VRAM is needed to for training Hyperscat.

Finally, in Table 5.36, we report the computational efficiency results for the split inverse and matching networks. Note that the size of the networks and required VRAM are approximately half that of the corresponding non-split networks. Thus, since splitting the networks only leads to a small degradation in performance (see Tables 5.34 and 5.35), splitting is a viable method for training if one does not have the computational resources needed to train the non-split networks. Note also that the combined MAC operations are noticeably less for the split networks than the non-split networks, which reduces inference time.

5.7 Conclusion

In this chapter, we have detailed Hyperscat, our pipeline for hyperspectral reconstruction which is based on the matching of discretized WST coefficients. While our initial numerical results were comparable to those of more complicated neural network architectures, our many experiments dedicated to optimizing our pipeline resulted in state of the art spatial and spectral reconstruction results. At the same time, we have demonstrated that our models are more computationally efficient in terms of size and VRAM requirements than these other networks.

This last feature of our pipelines is of particular value. While computational costs are important, memory usage has become the main bottleneck of AI and machine learning development due to memory's slower rate of improvement [55]. At the same time, there is a growing desire to implement and even train neural networks on small devices like smartphones [62]. In particular, the ICASSP 2024 Grand Challenge on Hyperspectral Skin Vision organizers' ultimate goal for the competition was for the creation of models that could fit on smartphones which would readily enable skin analysis through the reconstruction of hyperspectral images generated from pictures taken with a phone camera. Our pipelines' smaller memory footprints thus make them more practical and implementable for widespread generation and use of hyperspectral images.

Chapter 6: Properties of Generalized Fourier Scattering Transforms

6.1 Introduction

Fourier scattering transforms are known to exhibit properties such as Lipschitz continuity and stability to small deformations. What distinguishes them from general constructions of scattering transforms, however, is the fact that their K-th layer energies decay exponentially fast in K for every input function (see Theorem 2.2.18). This property is especially important for applications, as it means that only a few layers of scattering coefficients need to be calculated in order to capture the most important features of the data. While similar results have been proven for more general scattering transforms [52, 118, 122], these results only hold for specific subspaces of $L^2(\mathbb{R}^d)$ and hence do not fully preclude slow decay.

However, fast decay is not the only desirable property that an application might require. For example, we have shown in Chapter 5 that the approximate inverse of the discretized WST proposed by Mallat and Angles [14] allows us to move between different modalities of scattering coefficients in order to perform hyperspectral reconstruction. Despite this usefulness, questions about invertibility of scattering transforms have not seen much research outside of the discrete wavelet case. In particular, there has not been study on whether constructions of scattering transforms exist which are both invertible (in the discrete or continuous setting) and also exhibit fast energy decay for all inputs.

In this chapter, we present our new construction of generalized Fourier scattering transforms

(GFSTs). In Section 6.2, we prove that such generalizations have similar properties to FSTs, including exponentially fast energy decay, and we show that these transforms are invertible under an additional assumption. In Section 6.3, we study questions on the stability of the inverse, and we show that, if a scattering transform (and in particular a GFST) is invertible, then no global stability bound can exist. Finally, in Section 6.4, we discuss our initial numerical experiments for inverting two different discretized GFSTs.

6.2 Generalized Fourier Scattering Transforms

When considering how to invert the Fourier scattering transform on $L^2(\mathbb{R}^d)$, the first thing one might consider is the 0th layer output. For a function f, recall that the output of the 0th layer of an FST is simply $f * g_0$, where g_0 is the output generating function of the uniform covering frame $\mathcal{G}_F$. Given that $\widehat{f * g_0} = \widehat{f} \cdot \widehat{g_0}$, we would be easily able to reconstruct $\widehat{f}$, and hence f, as long as $\widehat{g_0}$ never vanishes. However, if $\mathcal{G}_F$ is a uniform covering frame, this is in fact not possible, as each g_λ and g_0 are bandlimited by assumption.

In this section, we present our construction of a generalization of uniform covering frames and Fourier scattering transforms which allows for the use of non-bandlimited convolutional filters. We prove that these generalized Fourier scattering transforms satisfy similar properties to the standard FST, and in particular that the layer-wise energy decays exponentially fast. Moreover, we consider a particular example where the semi-discrete frame is a semi-discrete Gabor frame generated by a Gaussian.

6.2.1 ε-Uniform Covering Frames

Recall that if $f \in L^2(\mathbb{R}^d)$ and $R > 0$ and $\epsilon \in [0, 1)$ are constants, we say that f is (ε, R)-bandlimited if $(1-\varepsilon)\|f\|_{L^2} \leq \|\widehat{f}\|_{L^2(Q_R(0))}$, or equivalently, if $\varepsilon\|f\|_{L^2} \geq \|\widehat{f}\|_{L^2(\mathbb{R}^d \setminus Q_R(0))}$. In essence, this definition states that most of $\widehat{f}$ lies in the cube $Q_R(0)$ except for a small portion that contributes at most ε percentage of the total energy of f. With this notion in mind, we define ε-uniform covering Bessel sequences and frames.

Definition 6.2.1. *Let $\varepsilon \in (0, 1]$, and let $\Lambda_{F,\varepsilon}$ be an indexing set such that $0 \notin \Lambda_{F,\varepsilon}$. Then, we say that the collection of filters $\mathcal{G}_{F,\varepsilon} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_{F,\varepsilon}} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ is an ε-uniform covering Bessel sequence if it satisfies the following properties:*

1. *Assumptions on g_0 and g_λ. $\widehat{g_0}$ is nonzero on a neighborhood of the origin and $0 < |\widehat{g_0}(0)| \leq 1$. For all $\lambda \in \Lambda_{F,\varepsilon}$, supp$(\widehat{g_\lambda})$ is connected.*

2. *ε-Uniform covering property. For every $R > 0$, there exists $N = N(R) \in \mathbb{N}$ and points $\left\{\xi_{\lambda,n} \in \mathbb{R}^d : \lambda \in \Lambda_{F,\varepsilon}, n \in \{1, \ldots, N\}\right\}$ such that*

$$\sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})}(\xi) \leq 1 - \varepsilon \text{ for all } \xi \in \mathbb{R}^d. \tag{6.1}$$

3. *Bessel sequence condition. For all $\xi \in \mathbb{R}^d$,*

$$|\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \leq 1. \tag{6.2}$$

If the third condition is replaced by the frame condition that there exists some $0 < A \leq 1$ such that,

for all $\xi \in \mathbb{R}^d$,

$$A \leq |\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \leq 1, \tag{6.3}$$

then we say that $\mathcal{G}_{F,\varepsilon}$ *is an* ε*-uniform covering frame.*

The main difference between this definition and the definition for uniform covering frames (Definition 2.2.8) is in part (2) of the respective definitions. For uniform covering frames, the uniform covering property states that each $\text{supp}(\widehat{g_\lambda})$ is contained in the union of N cubes of side length $2R$. In contrast, the ε-uniform covering property requires that each $\text{supp}(\widehat{g_\lambda})$ is *mostly* contained in these cubes, and the sum of the what's left outside of these respective cubes of the $|\widehat{g_\lambda}|^2$ terms is small. Nevertheless, ε-uniform covering frames are in fact a generalization of uniform covering frames, as shown by the next proposition.

Proposition 6.2.2. *Let* $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. *Suppose that* g_0 *is bandlimited and* $|\widehat{g_0}(0)| = 1$. *Then,* $\mathcal{G}$ *is a uniform covering frame if and only if* $\mathcal{G}$ *is a* 1*-uniform covering frame with lower frame bound* 1.

Proof. ($\implies$) Suppose $\mathcal{G}$ is a uniform covering frame. Clearly, we only need to verify the 1-uniform covering property to show that $\mathcal{G}$ is a 1-uniform covering frame. Since $\mathcal{G}$ is a uniform covering frame, for every $R > 0$, there exists $N = N(R) \in \mathbb{N}$ and points $\{\xi_{\lambda,n} \in \mathbb{R}^d : \lambda \in \Lambda, n \in \{1, \ldots, n\}\}$ such that, for every $\lambda \in \Lambda$, $\text{supp}(\widehat{g_\lambda}) \subseteq \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})$. But, this clearly implies that $|\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})}(\xi) = 0$ for all $\xi \in \mathbb{R}^d$, as the supports of these two functions are disjoint. Since λ was arbitrary, this implies that

$$\sum_{\lambda \in \Lambda} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})}(\xi) = 0 = 1 - 1 \text{ for all } \xi \in \mathbb{R}^d.$$

That is, $\mathcal{G}$ satisfies the 1-uniform covering property and is hence a 1-uniform covering frame with lower frame bound 1.

($\Longleftarrow$) Suppose that $\mathcal{G}$ is a 1-uniform covering frame with lower frame bound 1. Then, for every $R > 0$, there exists $N = N(R) \in \mathbb{N}$ and points $\{\xi_{\lambda,n} \in \mathbb{R}^d : \lambda \in \Lambda, n \in \{1, \ldots, n\}\}$ such that

$$\sum_{\lambda \in \Lambda} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})}(\xi) = 1 - 1 = 0 \text{ for all } \xi \in \mathbb{R}^d.$$

Since the terms of the sum are non-negative, this implies that $|\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})}(\xi) = 0$ for all $\xi \in \mathbb{R}^d$, and so we must have that $\text{supp}(\widehat{g_\lambda}) \subseteq \bigcup_{n=1}^{N} Q_R(\xi_{\lambda,n})$ for all $\lambda \in \Lambda$. Thus, every g_λ is bandlimited, and we have that $\mathcal{G}$ satisfies the uniform covering property. Thus, $\mathcal{G}$ is a uniform covering frame. $\square$

Three other things should be noted about Definition 6.2.1. First, when comparing this definition to part 1 of Definition 2.2.8 for uniform covering frames, we see that the corresponding part 1 removes the requirement that the g_0, g_λ be bandlimited. In particular, this allows for the convolutional filters to be non-zero everywhere in frequency space (which will allow for an easy inversion of the scattering transform via standard Fourier transform methods), with the constraint that, outside of some region, each filter has very small frequency values that satisfy Equation 6.1. Second, similar to the case of uniform covering frames (see Remark 2.2.9), a sufficient (though not necessary) condition for part 2 of Definition 6.2.1 is that there exists some fixed $R > 0$ and $\{\xi_\lambda \in \mathbb{R}^d : \lambda \in \Lambda_{F,\varepsilon}\}$ such that

$$\sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\xi_\lambda)}(\xi) \leq 1 - \varepsilon \text{ for all } \xi \in \mathbb{R}^d.$$

Finally, note that if $\mathcal{G}_{F,\varepsilon}$ is an ε-uniform covering frame for some $\varepsilon \in (0, 1]$, then $\mathcal{G}_{F,\varepsilon}$ is an ε'-uniform covering frame for all $0 < \varepsilon' \leq \varepsilon$.

Before we proceed to our generalization of the Fourier scattering transform, we will consider a class of examples of ε-uniform covering frames in small dimensions constructed from frequency shifts of Gaussians.

Example 6.2.3. *Let $a > 0$ be a constant, and let $\Lambda_{F,\varepsilon} = \frac{1}{\sqrt{a}}\left(\mathbb{Z}^d \setminus \{0\}\right)$. Next, define h_0 and h_λ for $\lambda \in \Lambda_{F,\varepsilon}$ by $\widehat{h_0}(\xi) \coloneqq \frac{1}{\pi^{d/4}} e^{-\frac{a|\xi|^2}{2}}$ and $\widehat{h_\lambda}(\xi) \coloneqq \widehat{h_0}(\xi - \lambda)$ (we are using h_0 and h_λ in place of g_0 and g_λ for the moment, as we shall obtain the latter functions by dividing the former functions by a constant which will be derived in the work below). Using standard Fourier transform results, it is easily seen that $h_0(x) = \pi^{d/4}\left(\frac{2}{a}\right)^{d/2} e^{-\frac{2\pi^2|x|^2}{a}}$ and $h_\lambda(x) = e^{2\pi i \lambda \cdot x} h_0(x)$ for $\lambda \in \Lambda_{F,\varepsilon}$.*

Next, consider

$$|\widehat{h_0}(\xi)|^2 + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{h_\lambda}(\xi)|^2 = \frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-a\left|\xi - \frac{k}{\sqrt{a}}\right|^2} = \frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-\left|\sqrt{a}\xi - k\right|^2}. \tag{6.4}$$

Define $\mathcal{H}(\xi) \coloneqq \frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-|\xi - k|^2}$. Using the well-known Poisson summation formula from harmonic analysis for the function $\mathcal{H}$ at the point $\sqrt{a}\xi$ (since Gaussians are Schwartz), we see that

$$\frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-\left|\sqrt{a}\xi - k\right|^2} = \sum_{k \in \mathbb{Z}^d} e^{-\pi^2|k|^2} e^{-2\pi i \sqrt{a} k \cdot \xi} = 1 + \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2|k|^2} e^{-2\pi i \sqrt{a} k \cdot \xi}. \tag{6.5}$$

Looking at the series on the right hand side, we can bound it above:

$$\left| \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2|k|^2} e^{-2\pi i \sqrt{a} k \cdot \xi} \right| \leq \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2|k|^2},$$

which in particular is achieved when $\xi = 0$. Note that the largest terms on the right hand side will have value $e^{-\pi^2} \approx 5 \times 10^{-5}$. Since the remaining terms decay extremely quickly, this sum remains

very small for small d, and in particular for $d \in \{1, 2, 3\}$. In such cases, we see that $\mathcal{H}(\sqrt{a}\xi)$ is nearly constant and extremely close to 1 for all $\xi \in \mathbb{R}^d$.

Define $\widetilde{C} = \sup_{\xi \in \mathbb{R}^d} |\mathcal{H}(\sqrt{a}\xi)| = \mathcal{H}(0) = 1 + \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2 |k|^2}$. *Define* $g_0(x) = (\widetilde{C})^{-1} h_0(x)$ *and* $g_\lambda(x) = (\widetilde{C})^{-1} h_\lambda(x)$ *for all* $\lambda \in \Lambda_{F,\varepsilon}$. *Combining this with Equations 6.4 and 6.5, we thus see that*

$$A \coloneqq \frac{1}{\widetilde{C}} \left(1 - \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2 |k|^2} \right) \leq |\widehat{g_0}(\xi)|^2 + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \leq 1, \tag{6.6}$$

with $0 < A < 1$ *but where* A *is still very close to* 1.

This shows that the collection $\{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_{F,\varepsilon}}$ *satisfies the frame condition in small dimensions. To show that it is an* ε*-uniform covering frame, we thus need to verify that it satisfies the* ε*-uniform covering property. Let* $R = \frac{1}{2\sqrt{a}}$, *and consider*

$$\begin{aligned} |\widehat{g_0}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(0)}(\xi) + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\lambda)}(\xi) &= \frac{1}{\widetilde{C} \pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-|\sqrt{a}\xi - k|^2} \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\frac{k}{\sqrt{a}})}(\xi) \\ &= \frac{1}{\widetilde{C} \pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-|\sqrt{a}\xi - k|^2} \mathbb{1}_{\mathbb{R}^d \setminus Q_{1/2}(k)}(\sqrt{a}\xi). \end{aligned}$$

Define $\mathscr{H}(\xi) = (\widetilde{C})^{-1} \frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d} e^{-|\xi - k|^2} \mathbb{1}_{\mathbb{R}^d \setminus Q_{1/2}(k)}(\xi)$, *so that*

$$|\widehat{g_0}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(0)}(\xi) + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\lambda)}(\xi) = \mathscr{H}(\sqrt{a}\xi). \tag{6.7}$$

If we can show that there exists $\varepsilon \in (0, 1]$ *such that* $\mathscr{H}(\xi) \leq 1 - \varepsilon$ *for all* $\xi \in \mathbb{R}^d$, *then we will have that*

$$\sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\lambda)}(\xi) \leq |\widehat{g_0}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(0)}(\xi) + \sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus Q_R(\lambda)}(\xi) \leq 1 - \varepsilon, \tag{6.8}$$

thus proving the ε-uniform covering property.

Notice that $\mathscr{H}$ is 1-periodic in each dimension. Thus, we only need to verify that $\mathscr{H}(\xi) \leq 1 - \varepsilon$ for some $\varepsilon \in (0, 1]$ for all $\xi \in Q_{1/2}(0)$. Notice that, if $\xi \in Q_{1/2}(0)$,

$$\widetilde{C}\mathscr{H}(\xi) = \frac{1}{\pi^{d/2}} \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-|\xi - k|^2} = \left(1 - \frac{1}{\pi^{d/2}} e^{-|\xi|^2}\right) + \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2 |k|^2} e^{-2\pi i k \cdot x},$$

by the Poisson summation formula. Bounding this on $Q_{1/2}(0)$, we see that

$$\widetilde{C}\mathscr{H}(\xi) \leq \left(1 - \frac{1}{\pi^{d/2}} e^{-d/4}\right) + \sum_{k \in \mathbb{Z}^d \setminus \{0\}} e^{-\pi^2 |k|^2}, \tag{6.9}$$

which will be strictly less than 1 for all sufficiently small d (in particular, for $d = 1, 2, 3$). Combining, we see that $\mathcal{G}_{F,\varepsilon} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_{F,\varepsilon}}$ is an ε-uniform covering frame for all sufficiently small d and for some for some $\varepsilon \in (0, 1]$ (depending on d).

Next, we generalize the definition of Fourier scattering transforms (cf. Definition 2.2.10).

Definition 6.2.4. *Let $\varepsilon \in (0, 1]$ and let $\mathcal{G}_{F,\varepsilon}$ be an ε-uniform covering Bessel sequence with indexing set $\Lambda_{F,\varepsilon}$. Let $\mathcal{P}_{F,\varepsilon} = \bigcup_{k=0}^{\infty} \Lambda_{F,\varepsilon}^k$ be the set of all paths. The generalized Fourier scattering transform (GFST) $S[\mathcal{P}_{F,\varepsilon}] : L^2(\mathbb{R}^d) \to l^2 L^2(\mathcal{P}_{F,\varepsilon}; \mathbb{R}^d)$ is a scattering transform with semi-discrete Bessel sequence $\mathcal{G}_{F,\varepsilon}$ and scattering propagators $U_{F,\varepsilon}[p]$ for $p \in \mathcal{P}_{F,\varepsilon}$.*

In particular, we get generalized Fourier scattering transforms with filters which are derived from frequency shifts of Gaussians in dimensions $d = 1, 2, 3$, as shown in Example 6.2.3.

6.2.2 Properties of Generalized Fourier Scattering Transforms

As Generalized Fourier scattering transforms are themselves scattering transforms, they inherit the same properties that scattering transforms have. In particular, GFSTs are nonlinear, bounded operators (Corollary 2.2.14) which are also Lipschitz continuous (Theorem 2.2.12) and stable to small deformations of the input (Theorem 2.2.15). However, as GFSTs are generalizations of FSTs, the question remains on whether GFSTs also have exponential decay of the K-th layer energies as $K \to \infty$, cf. Theorem 2.2.18. As the next theorem will demonstrate, GFSTs do in fact have this property, though the decay is in general, while still exponential, not as fast as in the case of FSTs.

Before we state the theorem, however, we note that many other theorems about the decay rates of the K-th layer energies of general constructions of scattering transforms have been previously proved. In [118], it was shown that the energy can decay exponentially fast for semi-discrete wavelet frames in one dimension, while it was shown in [122] that the rate can be polynomially fast (and exponential in dimension 1) for more general classes of frames, as long as the frequency supports of the frame elements can be rotated so that they can lie in the principal octant $\{\xi \in \mathbb{R}^d : \xi_i \geq 0 \text{ for all } i\}$. In [52], it was shown that, in general, energy decay for a wide class of frames (including wavelet frames) can be arbitrarily slow, but they also demonstrated exponential decay of K-th layer energies in arbitrary dimensions, as long as the same octant condition from [122] was met as well as the additional assumption that the frame elements are bandlimited.

Our result differs from these in three key respects. First, the exponential decay from the 3 aforementioned papers only occurs for specific subspaces of $L^2(\mathbb{R}^d)$. In contrast, our result shows exponential decay for all $f \in L^2(\mathbb{R}^d)$. Second, the other three papers all put restraints on the frequency supports of their convolutional filters. Our results, however, can hold for filters whose Fourier trans-

forms are non-zero on all of $\mathbb{R}^d$, though constraints still exist for the whole collection of filters in the form of the ε-uniform covering property. Finally, the proofs of these 3 previous results all mainly consist of bounding the K-th layer energies of the input $f \in L^2(\mathbb{R}^d)$ by a sequence of integral operators of $\widehat{f}$. Our proof, however, is based off ideas from the proof of Theorem 2.2.18 from [44], where the uniform (or in our case ε-uniform) covering property is exploited to bound the energy of f in terms of an exponentially decreasing sequence times $\|f\|_{L^2}^2$.

Theorem 6.2.5. *Let $\varepsilon \in (0, 1]$, and let $S[\mathcal{P}_{F,\varepsilon}]$ be a generalized Fourier scattering transform with ε-uniform covering frame $\mathcal{G}_{F,\varepsilon}$. Then, there exists a sequence of constants $C_K > 0$, $K \in \mathbb{N}$ (depending on $\mathcal{G}_{F,\varepsilon}$ and ε) such that C_K decays to 0 exponentially fast as $K \to \infty$ (i.e., there exists some constants $C, a > 0$ such that $C_K \leq Ce^{-aK}$ for all K) and satisfies, for all $f \in L^2(\mathbb{R}^d)$,*

$$\sum_{p \in \Lambda_{F,\varepsilon}^K} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 \leq C_K \|f\|_{L^2}^2. \tag{6.10}$$

Proof. Since $0 < |\widehat{g_0}(0)| \leq 1$, we have by a rescaling of the function in Proposition 3.2 from [44] that there exists a non-negative function φ such that $\widehat{\varphi}$ is continuous, decreasing along each Euclidean axis, $|\widehat{\varphi}(0)| > 0$, and $|\widehat{\varphi}(\xi)| \leq |\widehat{g_0}(\xi)|$ for all $\xi \in \mathbb{R}^d$. Hence, there exists some $R = R_\varphi > 0$ and some constant $C = C_\varphi \in (0, 1)$ such that $|\widehat{\varphi}(\xi)|^2 \geq C$ for all $\xi \in Q_R(0)$. By the ε-uniform covering frame, there exists $N = N(R) \in \mathbb{N}$ and points $\{\xi_{\lambda,n} : \lambda \in \Lambda_{F,\varepsilon}, n \in \{1, \ldots, N\}\}$ such that, for all $\xi \in \mathbb{R}^d$,

$$\sum_{\lambda \in \Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{\mathbb{R}^d \setminus \bigcup_{n=1}^N Q_R(\xi_{\lambda,n})}(\xi) \leq 1 - \varepsilon. \tag{6.11}$$

For notational simplicity, let $A(\lambda) = \mathbb{R}^d \setminus \bigcup_{n=1}^N Q_R(\xi_{\lambda,n})$ for each $\lambda \in \Lambda_{F,\varepsilon}$. Let $1 \leq k \leq K$ and

$q \in \Lambda_{F,\varepsilon}^{k-1}$. Using the Plancherel Theorem, we see that for every $\lambda \in \Lambda_{F,\varepsilon}$

$$\begin{aligned}\|U_{F,\varepsilon}[q]f * g_\lambda\|_{L^2}^2 &= \int_{\mathbb{R}^d} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi \\ \leq \sum_{n=1}^{N} \int_{Q_R(\xi_{\lambda,n})} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi &+ \int_{A(\lambda)} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi.\end{aligned}$$

For the first term, we follow the argument for the proof of the normal Fourier scattering transform decay (see the proof of Proposition 3.3 in [44], and cf. the proof of Theorem 4.4.3): since $|\widehat{\varphi}(\xi - \xi_{\lambda,n})|^2 \geq C$ for all $\xi \in Q_R(\xi_{\lambda,n})$ and by the Plancherel Theorem, we have that

$$\begin{aligned}\sum_{n=1}^{N} \int_{Q_R(\xi_{\lambda,n})} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 |\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi &\leq \frac{1}{C} \sum_{n=1}^{N} \int_{Q_R(\xi_{\lambda,n})} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 |\widehat{g_\lambda}(\xi)|^2 |\widehat{\varphi}(\xi - \xi_{\lambda,n})|^2 \mathrm{d}\xi \\ &= \frac{1}{C} \sum_{n=1}^{N} \|U_{F,\varepsilon}[q]f * g_\lambda * M_{\xi_{\lambda,n}}\varphi\|_{L^2}^2.\end{aligned}$$

Given that $\varphi \geq 0$, we can pull the modulus inside the convolution to see that

$$\frac{1}{C} \sum_{n=1}^{N} \|U_{F,\varepsilon}[q]f * g_\lambda * M_{\xi_{\lambda,n}}\varphi\|_{L^2}^2 \leq \frac{1}{C} \sum_{n=1}^{N} \||U_{F,\varepsilon}[q]f * g_\lambda| * \varphi\|_{L^2}^2 = \frac{N}{C} \||U_{F,\varepsilon}[q]f * g_\lambda| * \varphi\|_{L^2}^2$$

Given that $|\widehat{\varphi}(\xi)| \leq |\widehat{g_0}(\xi)|$ everywhere, we can apply the Plancherel theorem again to see that

$$\frac{N}{C} \||U_{F,\varepsilon}[q]f * g_\lambda| * \varphi\|_{L^2}^2 \leq \frac{N}{C} \||U_{F,\varepsilon}[q]f * g_\lambda| * g_0\|_{L^2}^2 .$$

Thus, we have

$$\|U_{F,\varepsilon}[q]f * g_\lambda\|_{L^2}^2 \leq \frac{N}{C}\||U_{F,\varepsilon}[q]f * g_\lambda| * g_0\|_{L^2}^2 + \int\limits_{A(\lambda)} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2|\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi.$$

Summing over all $\lambda \in \Lambda_{F,\varepsilon}$ and using the ε-uniform covering property, notice that

$$\begin{aligned}\sum_{\lambda\in\Lambda_{F,\varepsilon}} \int\limits_{A(\lambda)} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2|\widehat{g_\lambda}(\xi)|^2 \mathrm{d}\xi &= \int\limits_{\mathbb{R}^d} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 \sum_{\lambda\in\Lambda_{F,\varepsilon}} |\widehat{g_\lambda}(\xi)|^2 \mathbb{1}_{A(\lambda)}(\xi)\mathrm{d}\xi \\ &\leq (1-\varepsilon)\int\limits_{\mathbb{R}^d} |\widehat{U_{F,\varepsilon}[q]f}(\xi)|^2 \mathrm{d}\xi.\end{aligned}$$

Hence, we see that

$$\sum_{\lambda\in\Lambda_{F,\varepsilon}} \|U_{F,\varepsilon}[q]f * g_\lambda\|_{L^2}^2 \leq \sum_{\lambda\in\Lambda_{F,\varepsilon}} \left(\frac{N}{C}\||U_{F,\varepsilon}[q]f * g_\lambda| * g_0\|_{L^2}^2\right) + (1-\varepsilon)\|U_{F,\varepsilon}[q]f\|_{L^2}^2.$$

Summing over $q \in \Lambda_{F,\varepsilon}^{k-1}$ and noting that every $p \in \Lambda_{F,\varepsilon}^k$ can be written as $p = (q, \lambda)$ for $q \in \Lambda_{F,\varepsilon}^{k-1}, \lambda \in \Lambda_{F,\varepsilon}$, we see that

$$\frac{C}{N}\sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 \leq \sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f * g_0\|_{L^2}^2 + \frac{(1-\varepsilon)C}{N}\sum_{q\in\Lambda_{F,\varepsilon}^{k-1}} \|U_{F,\varepsilon}[q]f\|_{L^2}^2.$$

Using the uniform covering Bessel sequence condition, we have that

$$\sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f * g_0\|_{L^2}^2 \leq \sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 - \sum_{p\in\Lambda_{F,\varepsilon}^{k+1}} \|U_{F,\varepsilon}[p]f\|_{L^2}^2. \tag{6.12}$$

Combining these inequalities, we have that

$$\sum_{p\in\Lambda_{F,\varepsilon}^{k+1}} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 \leq \left(1-\frac{C}{N}\right)\sum_{p\in\Lambda_{F,\varepsilon}^{k}} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 + \frac{(1-\varepsilon)C}{N}\sum_{q\in\Lambda_{F,\varepsilon}^{k-1}} \|U_{F,\varepsilon}[q]f\|_{L^2}^2. \tag{6.13}$$

For notational simplicity, let $a_k = \sum_{\lambda\in\Lambda_{F,\varepsilon}^{k}} \|U_{F,\varepsilon}[p]f\|_{L^2}^2$. We can then rewrite Inequality 6.13 as

$$a_{k+1} \leq \left(1-\frac{C}{N}\right)a_k + \frac{(1-\varepsilon)C}{N}a_{k-1}$$

for all $k \in \mathbb{N}$, $k \geq 1$. By Inequality 6.12, we can also conclude that $a_1 \leq a_0 = \|f\|_{L^2}^2$. Now, suppose that $\{b_k : k = 0, 1, \ldots\}$ is a sequence recursively defined by $b_0 = b_1 = \|f\|_{L^2}^2$ and

$$b_k = \left(1-\frac{C}{N}\right)b_{k-1} + \frac{(1-\varepsilon)C}{N}b_{k-2} \tag{6.14}$$

for $k \geq 2$. By an inductive argument, we clearly have that $0 \leq a_k \leq b_k$ for all k. Thus, finding a general formula for b_k will give us a bound for a_k.

Clearly, the (complex) vector space of solutions to the recursion Equation 6.14 has dimension at most 2. Suppose that $b_k = r^k$ is a solution for some $r \in \mathbb{C} \setminus \{0\}$. Then, for $k \geq 2$, Equation 6.14 can be rewritten as

$$r^k = \left(1-\frac{C}{N}\right)r^{k-1} + \frac{(1-\varepsilon)C}{N}r^{k-2}.$$

Since $r \neq 0$, we can divide out by r^{k-2} and we are left with

$$r^2 = \left(1-\frac{C}{N}\right)r + \frac{(1-\varepsilon)C}{N}.$$

By the quadratic formula, the solutions to this quadratic equation are

$$r_1 = \frac{(1-\frac{C}{N}) + \sqrt{(1-\frac{C}{N})^2 + 4(1-\varepsilon)\frac{C}{N}}}{2} \tag{6.15}$$

and

$$r_2 = \frac{(1-\frac{C}{N}) - \sqrt{(1-\frac{C}{N})^2 + 4(1-\varepsilon)\frac{C}{N}}}{2}. \tag{6.16}$$

We claim that r_1, r_2 are real valued with $-1 < r_2 \leq 0 \leq r_1 < 1$ and that $|r_2| < |r_1|$.

Since $C \in (0,1), \varepsilon \in (0,1]$, and $N \geq 1$, we clearly have that the discriminant $\widetilde{D} \coloneqq (1 - \frac{C}{N})^2 + 4(1-\varepsilon)\frac{C}{N}$ is positive. Thus, $r_1, r_2 \in \mathbb{R}$. Similarly, it is clear that $r_1 \geq 0$. Also, since $\sqrt{\widetilde{D}} \geq \sqrt{(1-\frac{C}{N})^2} = (1-\frac{C}{N})$, we see that $r_2 \leq 0$. Since r_1 is the sum of two positive numbers and r_2 is the different of those same two positive numbers, it is also clear that $|r_2| < |r_1|$.

In order for $r_1 < 1$ to be true, we would need

$$\sqrt{\left(1-\frac{C}{N}\right)^2 + 4(1-\varepsilon)\frac{C}{N}} < 2 - \left(1-\frac{C}{N}\right).$$

Squaring both sides, we see this requires

$$\left(1-\frac{C}{N}\right)^2 + 4(1-\varepsilon)\frac{C}{N} < 4 - 4\left(1-\frac{C}{N}\right) + \left(1-\frac{C}{N}\right)^2.$$

Rearranging, this requires

$$1 > \frac{(1-\varepsilon)C}{N} + \left(1-\frac{C}{N}\right) = 1 - \frac{\varepsilon C}{N}.$$

Since this is clearly true, we see that $r_1 < 1$. By a similar argument, we have $-1 < r_2$.

Clearly, the sequences r_1^k and r_2^k are linearly independent over $\mathbb{C}$. Hence, they must be the

generating solutions, and so the general solution to Equation 6.14 is $b_k = ar_1^k + br_2^k$ for some constants a, b. Plugging in the initial conditions, we see that $\|f\|_{L^2}^2 = a + b$ and $\|f\|_{L^2}^2 = ar_1 + br_2$. Solving for a and b, we see that $a = \frac{1-r_2}{\sqrt{\widetilde{D}}}\|f\|_{L^2}^2$ and $b = \frac{r_1-1}{\sqrt{\widetilde{D}}}\|f\|_{L^2}^2$. Thus, we have that

$$b_k = \left(\frac{(1-r_2)}{\sqrt{\widetilde{D}}} r_1^k + \frac{(r_1-1)}{\sqrt{\widetilde{D}}} r_2^k \right) \|f\|_{L^2}^2 \tag{6.17}$$

for all k, where $\widetilde{D} = \left(1 - \frac{C}{N}\right)^2 + 4(1-\varepsilon)\frac{C}{N}$ and r_1 and r_2 are defined by Equations 6.15 and 6.16, respectively. It is clear that $b_k \geq 0$, and since $|r_2| < |r_1| < 1$, we have that b_k decays exponentially fast to 0 as $k \to \infty$. Taking $C_K = b_K$ for all K completes the proof. □

Remark 6.2.6. *Consider the case where $\varepsilon = 1$. Then, $\tilde{D} = \left(1 - \frac{C}{N}\right)^2$ so $r_1 = \left(1 - \frac{C}{N}\right)$ and $r_2 = 0$. Moreover, $C_K = \left(1 - \frac{C}{N}\right)^{K-1}$, which is precisely the bounding sequence in Theorem 2.2.18. Thus, Theorem 6.2.5 is in fact a generalization of Theorem 2.2.18. Moreover, combining the proofs of Theorems 6.2.5 and 4.4.3, we can prove that the results of the former theorem still hold if we replace $f \in L^2(\mathbb{R}^d)$ with a mean square continuous, strictly stationary process $X(\omega, t)$.*

With the preceding energy decay results, we can show that generalized Fourier scattering transforms are bounded below in certain cases.

Corollary 6.2.7. *Suppose that $\mathcal{G}_{F,\varepsilon}$ is a ε-uniform covering frame, with lower frame bound $0 < A \leq 1$.*

(1) Suppose that $A < 1$ and that there exists some $K \in \mathbb{N}$ such that $C_K < A^K$, where $\{C_K : K = 0, 1, \ldots\}$ is the sequence from Theorem 6.2.5. Then, for all $f \in L^2(\mathbb{R}^d)$, we have

$$(A^K - C_K)\|f\|_{L^2}^2 \leq \|S[\mathcal{P}_{\mathcal{F},\varepsilon}]f\|_{l^2 L^2}^2. \tag{6.18}$$

(2) Suppose that $A = 1$. Then, for all $f \in L^2(\mathbb{R}^d)$,

$$\|f\|_{L^2}^2 = \|S[\mathcal{P}_{\mathcal{F},\varepsilon}]f\|_{l^2L^2}^2. \tag{6.19}$$

Proof. Proof of Part (1): By Proposition 1 in [122], we have that

$$A^K\|f\|_{L^2}^2 \leq \sum_{k=0}^{K-1}\sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f * g_0\|_{L^2}^2 + \sum_{p\in\Lambda_{F,\varepsilon}^K} \|U_{F,\varepsilon}[p]f\|_{L^2}^2. \tag{6.20}$$

By Theorem 6.2.5, we have that $\sum_{p\in\Lambda_{F,\varepsilon}^K} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 \leq C_K\|f\|_{L^2}^2$. Rearranging, we thus get that

$$(A^K - C_K)\|f\|_{L^2}^2 \leq \sum_{k=0}^{K-1}\sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f * g_0\|_{L^2}^2 \leq \|S[\mathcal{P}_{F,\varepsilon}]f\|_{l^2L^2}^2.$$

Proof of Part (2): Applying Proposition 1 in [122] again, we have that

$$\|f\|_{L^2}^2 = \sum_{k=0}^{K-1}\sum_{p\in\Lambda_{F,\varepsilon}^k} \|U_{F,\varepsilon}[p]f * g_0\|_{L^2}^2 + \sum_{p\in\Lambda_{F,\varepsilon}^K} \|U_{F,\varepsilon}[p]f\|_{L^2}^2, \tag{6.21}$$

for all $f \in L^2(\mathbb{R}^d)$ and all K. Since $\sum_{p\in\Lambda_{F,\varepsilon}^K} \|U_{F,\varepsilon}[p]f\|_{L^2}^2 \to 0$ as $K \to \infty$ by Theorem 6.2.5, we have by taking the limit as $K \to \infty$ in Equation 6.21 that

$$\|f\|_{L^2}^2 = \|S[\mathcal{P}_{\mathcal{F},\varepsilon}]f\|_{l^2L^2}^2.$$

□

Remark 6.2.8. *In order for $A^K > C_K$ for some K to be satisfied, it is enough that C_K decays faster*

than A^K. Given that r_1 (as defined in the proof of Theorem 6.2.5) determines an upper bound for the decay of C_K (since $|r_1| > |r_2|$), a sufficient condition is that $r_1 < A$.

While the aforementioned properties are generalizations of corresponding properties for normal FSTs, the lack of requirement that g_0 be bandlimited will allow us to construct GFSTs which are invertible.

Theorem 6.2.9. *Suppose that $\mathcal{G}_{F,\varepsilon} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_{F,\varepsilon}}$ is an ε-uniform covering frame. Suppose further that $\widehat{g_0} \neq 0$ m-a.e. Then, $S[\mathcal{P}_{F,\varepsilon}]$ is an invertible operator on its range.*

Proof. Let $f \in L^2(\mathbb{R}^d)$, and consider $S[\mathcal{P}_{\mathcal{F},\varepsilon}]f = \{U_{F,\varepsilon}[p]f * g_0 : p \in \mathcal{P}_{F,\varepsilon}\}$. We will show that we can reconstruct f from $S[\mathcal{P}_{F,\varepsilon}]f$. The 0-th layer output of the transform is $U_{F,\varepsilon}[\varnothing]f * g_0 = f * g_0$. Taking the Fourier transform of the 0-th layer, we get $\widehat{f * g_0} = \widehat{f}\widehat{g_0}$. Since $\widehat{g_0} \neq 0$ m-a.e., we can divide m-a.e. by $\widehat{g_0}$ to recover $\widehat{f}(\xi)$ m-a.e. Finally, by taking the inverse Fourier transform, we recover f. That is, this scattering transform is invertible. □

Before ending this section, we return to the example of an ε-uniform covering frame which is generated by frequency shifts of a Gaussian.

Example 6.2.10. *Consider the filters from Example 6.2.3. Since g_0 is a Gaussian, it satisfies the condition of Theorem 6.2.9, and so $S[\mathcal{P}_{F,\varepsilon}]$ is an invertible operator. Let's numerically calculate the constants ε, A and the sequence C_K for dimensions $d = 1, 2$.*

In dimension $d = 1$, we have that $0.00010 \leq \sum_{k \in \mathbb{Z}\setminus\{0\}} e^{-\pi^2|k|^2} \leq 0.00011$, so $A \geq \frac{1-0.00011}{1+0.00011} \geq 0.99978$. Using Inequality 6.9 and the fact that $\frac{1}{\sqrt{\pi}}e^{-1/4} \geq 0.43939$, we have that

$$\mathscr{H}(\xi) \leq \frac{1}{1.00010}\left(1 - 0.43939 + 0.00011\right) \leq 0.56067.$$

Thus, for $d = 1$*, we have that* $\varepsilon = 1 - 0.56067 = 0.43933$*, and so the filters* $\{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda_{F,\varepsilon}}$ *are a* 0.43933*-uniform covering frame with lower frame bound* 0.99978*.*

Next, in the proof of Theorem 6.2.5, we take $\varphi = g_0$*,* $R = \frac{1}{2}$*, and* $C = 0.43934 \leq \frac{1}{\widehat{C}\sqrt{\pi}} e^{-1/2}$*, so that* $C \leq |\widehat{\varphi}(\xi)|$ *for all* $\xi \in Q_{1/2}(0)$*. Then, as this was the same* R *we used to find* ε*, we can take* $N = 1$ *for Equation 6.11, and hence* $\frac{C}{N} = 0.43934$*. Plugging these constants into Equations 6.15 and 6.16, we get that* $r_1 \approx 0.85033$ *and* $-0.28968 \leq r_2 \leq -0.28967$*. Combining, we get that*

$$C_K \approx (1.13128)(0.85033)^K - 0.13128(-0.28967)^K.$$

Given that we can choose any K *for which* $A^K > C_K$ *for Equation 6.18 to determine a lower bound for the scattering operator, we can take* $K = 100$ *and we get that, for all* $f \in L^2(\mathbb{R}^d)$*, the GFST has lower bound* 0.97823*.*

We can repeat these calculations for $d = 2$*. In that case, we get* $A \geq 0.99978$*,* $\varepsilon = 0.19302$*,* $C_K \approx (1.07291)(0.92409)^K - (0.07291)(-0.11712)^K$*, and a lower bound for the GFST of* 0.97783*.*

6.3 Stability of Inversion of Scattering Transforms

We have constructed examples of scattering transforms which have desirable properties, like exponentially fast layer-wise energy decay, and which are also invertible operators. With this, however, comes questions as to whether the inverse is stable. In other words, we are interested in whether an invertible scattering transform is globally or locally bi-Lipschitz, i.e. whether there exists a (global or local constant) $\gamma > 0$ such that

$$\gamma \|f - g\|_{L^2} \leq \|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{L^2 l^2}, \tag{6.22}$$

either for all $f, g \in L^2(\mathbb{R}^d)$ (in the global case) or just for all g in some neighborhood of f (in the local case). Such a result would also have implications for applications. For example, bi-Lipschitzity would ensure that the scattering transform embedding keeps inputs that were close in the original space still close in the embedding space while also ensuring that inputs that were far apart remained separated. Such properties help classifiers more easily distinguish between different classes.

In an attempt to begin to answer these questions for GFSTs, we shall return to the broader setting of scattering transforms with a semi-discrete frame $\mathcal{G}$. When attempting to find a global lower bi-Lipschitz bound, the first thing we might consider is whether scattering transforms can stably distinguish between functions and unimodular constants times those same functions. So, let $f \in L^2(\mathbb{R}^d)$, and let $\nu \in \mathbb{C}$ such that $|\nu| = 1$. Notice that, for all $\lambda \in \Lambda$, we have $(\nu f) * g_\lambda = \nu(f * g_\lambda)$, so $U[\lambda](\nu f) = |f * g_\lambda| = U[\lambda]f$. Thus, for all $p \in \mathcal{P}$ with $p \neq \varnothing$, we have $U[p](\nu f) = U[p]f$, and so

$$\|S[\mathcal{P}]f - S[\mathcal{P}](\nu f)\|_{l^2L^2} = \|f * g_0 - (\nu f) * g_0\|_{L^2} = |1 - \nu|\|f * g_0\|_{L^2}. \tag{6.23}$$

Similarly, we have $\|f - \nu f\|_{L^2} = |1 - \nu|\|f\|_{L^2}$, so in order for a global lower bi-Lipschitz bound to exist, there would need to exist a constant $\gamma > 0$ such that

$$\gamma\|f\|_{L^2} \leq \|f * g_0\|_{L^2},$$

for all $f \in L^2(\mathbb{R}^d)$, which is clearly impossible unless $\widehat{g_0}$ is bounded below uniformly by γ. Since $g_0 \in L^1(\mathbb{R}^d)$, we have $\widehat{g_0}(\xi) \to 0$ as $|\xi| \to \infty$ by the Riemann-Lebesgue lemma, and so no such global constant could exist.

While no global lower bound can exist with respect to the L^2 distance of f, g, one may ask

instead about whether lower bi-Lipschitz bounds exist with respect to a different norm. Note that, for each $\lambda \in \Lambda$, we have $U[\lambda]f(x) = |\langle f, T_x g_\lambda^* \rangle|$ (see Equation 2.8). Thus, the question of stably inverting the scattering transform looks similar to the question of recover f from the modulus of its frame coefficients, a problem known as *phase retrieval* [5]. In this problem, functions are only recovered (stably or not) up to a *unimodular constant*. That is, one considers the equivalence relation $f \sim g$ if there exists $\nu \in \mathbb{C}$, $|\nu| = 1$ such that $f = \nu g$. Questions of stability are then considered with respect to the quotient norm $\|f - g\|_{L^2/\sim} \coloneqq \min_{\substack{\nu \in \mathbb{C} \\ |\nu|=1}} \|f - \nu g\|_{L^2}$.

With this in mind, we shall look for bi-Lipschitz bounds with respect to the $L^2(\mathbb{R}^d)/\sim$ quotient norm in this section. We shall show that, similar to the phase retrieval problem [5], no global lower bi-Lipschitz bound can exist with respect to this norm for scattering transforms constructed from semi-discrete frames, and hence in particular that the GFST inversion can never be globally stable. However, we shall also demonstrate that lower bounds can exist for certain subspaces of $L^2(\mathbb{R}^d)$. Finally, we shall discuss possible avenues for studying the existence of local lower bi-Lipschitz bounds for both FSTs and GFSTs.

6.3.1 Global Bi-Lipschitz Bounds

In order to prove that no global lower bi-Lipschitz bound can exist, we follow the work of [5] which proved a similar result for frame phase retrieval in infinite-dimensional Banach spaces. We first need the following two lemmas.

Lemma 6.3.1. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda}$ be a semi-discrete Bessel sequence (with upper frame bound*

$B = 1$*), and let* $S[\mathcal{P}]$ *be it's corresponding scattering transform. Then, for all* $f, g \in L^2(\mathbb{R}^d)$*,*

$$\|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2L^2}^2 \leq \|f * g_0 - g * g_0\|_{L^2}^2 + \sum_{\lambda \in \Lambda} \||f * g_\lambda| - |g * g_\lambda|\|_{L^2}^2. \tag{6.24}$$

Proof. The proof of this lemma is similar to that of Theorem 2.2.12. Fix some $k \geq 1$, and let $p \in \Lambda^k$. By the reverse-triangle inequality and the Bessel sequence condition (Equation 2.4), we have that

$$\begin{aligned}
&\|U[p]f * g_0 - U[p]g * g_0\|_{L^2}^2 + \sum_{\lambda_{k+1} \in \Lambda} \|U[(p, \lambda_{k+1})]f - U[(p, \lambda_{k+1})]g\|_{L^2}^2 \\
\leq &\|U[p]f * g_0 - U[p]g * g_0\|_{L^2}^2 + \sum_{\lambda_{k+1} \in \Lambda} \|U[p]f * g_{\lambda_{k+1}} - U[p]g * g_{\lambda_{k+1}}\|_{L^2}^2 \\
\leq &\|U[p]f - U[p]g\|_{L^2}^2.
\end{aligned}$$

Summing over $p \in \Lambda^k$ and recalling that every $q \in \Lambda^{k+1}$ can be written as (p, λ_{k+1}) for some $p \in \Lambda^k$ and $\lambda_{k+1} \in \Lambda$, we can rewrite this inequality as

$$\sum_{p \in \Lambda^k} \|U[p]f * g_0 - U[p]g * g_0\|_{L^2}^2 + \sum_{q \in \Lambda^{k+1}} \|U[q]f - U[q]g\|_{L^2}^2 \leq \sum_{p \in \Lambda^k} \|U[p]f - U[p]g\|_{L^2}^2.$$

Summing the first term on the left hand side from $k = 1$ to K, we see that

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{p \in \Lambda^k} \|U[p]f * g_0 - U[p]g * g_0\|_{L^2}^2 \\
\leq &\sum_{k=1}^{K} \left(\sum_{p \in \Lambda^k} \|U[p]f - U[p]g\|_{L^2}^2 - \sum_{p \in \Lambda^{k+1}} \|U[p]f - U[p]g\|_{L^2}^2 \right) \\
= &\sum_{p \in \Lambda^1} \|U[p]f - U[p]g\|_{L^2}^2 - \sum_{p \in \Lambda^{K+1}} \|U[p]f - U[p]g\|_{L^2}^2 \leq \sum_{p \in \Lambda^1} \|U[p]f - U[p]g\|_{L^2}^2.
\end{aligned}$$

Since this is true for every K, we see that

$$\begin{aligned}\|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2L^2}^2 =& \|f * g_0 - g * g_0\|_{L^2}^2 + \sum_{k=1}^{\infty}\sum_{p\in\Lambda^k} \|U[p]f * g_0 - U[p]g * g_0\|_{L^2}^2 \\ \leq& \|f * g_0 - g * g_0\|_{L^2}^2 + \sum_{p\in\Lambda^1} \|U[p]f - U[p]g\|_{L^2}^2 \\ =& \|f * g_0 - g * g_0\|_{L^2}^2 + \sum_{\lambda\in\Lambda} \||f * g_\lambda| - |g * g_\lambda|\|_{L^2}^2,\end{aligned}$$

as claimed. □

Lemma 6.3.2. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda\in\Lambda}$ be a semi-discrete frame with lower frame bound $A > 0$ and upper frame bound 1. Suppose that there exists some $\gamma > 0$ such that, for all $f, g \in L^2(\mathbb{R}^d)$, we have*

$$\gamma\|f - g\|_{L^2/\sim} \leq \|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2L^2}. \tag{6.25}$$

Then, we must have $\gamma \leq 1$.

Proof. By Theorem 2.2.12, we have that $S[\mathcal{P}]$ is a Lipschitz operator with bound 1. Notice that, for all $\nu \in \mathbb{C}, |\nu| = 1$, we have $\|f - g\|_{L^2/\sim} = \|f - \nu g\|_{L^2/\sim}$. Applying Equation 6.25 and Theorem 2.2.12 for each ν, we thus have that

$$\gamma\|f - g\|_{L^2/\sim} = \gamma\|f - \nu g\|_{L^2/\sim} \leq \|S[\mathcal{P}]f - S[\mathcal{P}](\nu g)\|_{l^2L^2} \leq \|f - \nu g\|_{L^2}.$$

Since this is true for every ν, it is true if we take the minimum over all ν:

$$\gamma\|f - g\|_{L^2/\sim} \leq \min_{\substack{\nu\in\mathbb{C}\\|\nu|=1}} \|f - \nu g\|_{L^2} = \|f - g\|_{L^2/\sim}.$$

Since $f, g \in L^2(\mathbb{R}^d)$ were arbitrary, we must have $\gamma \leq 1$. $\square$

In order to prove that no global lower bi-Lipschitz bound exists, we first need to define the σ-strong complement property. Before we get to this definition, however, we need some setup. Equip $\Lambda \cup \{0\}$ with the discrete topology, and equip $\Lambda^* \coloneqq (\Lambda \cup \{0\}) \times \mathbb{R}^d$ with the product topology. Then, we have that $(\Lambda^*, \mathcal{B}(\Lambda^*), \widetilde{c} \times m)$ is a (σ-finite) measure space, where $\widetilde{c}$ is the counting measure. Note that the measurable sets are precisely of the form $\bigcup_{\lambda \in \Lambda \cup \{0\}} \{\lambda\} \times A_\lambda$, where $A_\lambda \in \mathcal{B}(\mathbb{R}^d)$, since $\Lambda \cup \{0\}$ is given the discrete topology.

Clearly, any measurable function $\mathscr{F} : \Lambda^* \to \mathbb{C}$ can be written as $\mathscr{F}(\lambda, x) = f_\lambda(x)$ for a sequence of $\mathcal{B}(\mathbb{R}^d)$-measurable functions $\{f_\lambda : \mathbb{R}^d \to \mathbb{C} : \lambda \in \Lambda \cup \{0\}\}$. Thus, for any measurable subset $\mathcal{S} = \bigcup_{\lambda \in \Lambda \cup \{0\}} \{\lambda\} \times A_\lambda \subseteq \Lambda^*$, we have that

$$\|\mathscr{F}\|^2_{L^2(S)} = \int_{\mathcal{S}} |\mathscr{F}(\lambda, x)|^2 \mathrm{d}(\widetilde{c} \times m)(x, \lambda) = \sum_{\lambda \in \Lambda \cup \{0\}} \int_{A_\lambda} |f_\lambda(x)|^2 \mathrm{d}m(x) = \|\{f_\lambda \mathbb{1}_{A_\lambda}\}\|^2_{l^2 L^2}. \quad (6.26)$$

Thus, we shall write $\|\{f_\lambda\}\|_{l^2 L^2(S)} \coloneqq \|\{f_\lambda \mathbb{1}_{A_\lambda}\}\|_{l^2 L^2}$. Note that we can rewrite the frame condition (Equation 2.5) as

$$A\|f\|^2_{L^2} \leq \|\{f * g_\lambda\}\|^2_{l^2 L^2(\Lambda^*)} \leq B\|f\|^2_{L^2} \text{ for all } f \in L^2(\mathbb{R}^d). \quad (6.27)$$

Now, we are ready to define the σ-strong complement property (cf. Definition 3.6 in [5]).

Definition 6.3.3. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. We say that $\mathcal{G}$ satisfies the σ-strong complement property (SCP) if there exists $\sigma \geq 0$ such that, for every measurable subset $\mathcal{S} \subseteq \Lambda \times \mathbb{R}^d \subseteq$*

Λ^*, *there exists constants* $A_{\mathcal{S}}, A_{\mathcal{T}} \geq 0$ *(where* $\mathcal{T} \coloneqq (\Lambda \times \mathbb{R}^d) \setminus \mathcal{S}$*) such that*

$$A_{\mathcal{S}}\|f\|_{L^2}^2 \leq \|f * g_0\|_{L^2}^2 + \|\{f * g_\lambda\}\|_{l^2L^2(\mathcal{S})}^2 \text{ for all } f \in L^2(\mathbb{R}^d) \tag{6.28}$$

and

$$A_{\mathcal{T}}\|f\|_{L^2}^2 \leq \|f * g_0\|_{L^2}^2 + \|\{f * g_\lambda\}\|_{l^2L^2(\mathcal{T})}^2 \text{ for all } f \in L^2(\mathbb{R}^d), \tag{6.29}$$

where $\max\{A_{\mathcal{S}}, A_{\mathcal{T}}\} \geq \sigma$*. We denote the largest* σ *for which* $\mathcal{G}$ *satisfies the* σ*-SCP property by* $\sigma_{\mathcal{G}}$*.*

Remark 6.3.4. *Note that this definition is not identical to Definition 3.6 in [5], as the latter would require that* $\mathcal{S} \subseteq \Lambda^*$ *be arbitrary. We needed to use a slightly different definition, however, due to the fact that we don't take the modulus of the convolution* $f * g_0$ *when calculating* $S[\mathcal{P}]f$*.*

Note that, if $\mathcal{G}$ has lower frame bound A and we choose $\mathcal{S} = \Lambda \times \mathbb{R}^d$, then $A_{\mathcal{S}} = A$ and $A_{\mathcal{T}} = 0$, and so we necessarily need $A \geq \sigma_{\mathcal{G}}$. We will now show that it is necessary that $\mathcal{G}$ be σ-SCP for some $\sigma > 0$ in order for a global bi-Lipschitz bound to exist, cf. Theorem 3.9 from [5].

Theorem 6.3.5. *Let* $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda \in \Lambda}$ *be a semi-discrete frame with lower frame bound* $A > 0$ *and upper frame bound* 1*. Suppose that there exists* $\gamma \geq 0$ *such that Equation 6.25 holds for all* $f, g \in L^2(\mathbb{R}^d)$*. Then, we must have*

$$\gamma^2 \leq \frac{8}{A}\sigma_{\mathcal{G}}. \tag{6.30}$$

Proof. We consider two cases. If $0 < \frac{1}{4}A \leq \sigma_{\mathcal{G}} \leq A$, then using Lemma 6.3.2, we have that

$$\gamma^2 \leq 1 = \frac{\sigma_{\mathcal{G}}}{\sigma_{\mathcal{G}}} \leq \sigma_{\mathcal{G}}\frac{4}{A} \leq \frac{8}{A}\sigma_{\mathcal{G}}.$$

Otherwise, suppose $\sigma_{\mathcal{G}} < \frac{1}{4}A$. Let $\sigma_{\mathcal{G}} < \sigma \leq \frac{A}{4}$. Since $\sigma_{\mathcal{G}}$ is the largest σ for which $\mathcal{G}$ satisfies

the σ-SCP property, there must exist a measurable subset $\mathcal{S} \subseteq \Lambda \times \mathbb{R}^d$ (with $\mathcal{T} \coloneqq (\Lambda \times \mathbb{R}^d) \setminus \mathcal{S}$) and two functions $u, v \in L^2(\mathbb{R}^d)$, $\|u\|_{L^2} = \|v\|_{L^2} = 1$ such that

$$\|u * g_0\|_{L^2}^2 + \|\{u * g_\lambda\}\|_{l^2 L^2(\mathcal{S})}^2, \|v * g_0\|_{L^2}^2 + \|\{v * g_\lambda\}\|_{l^2 L^2(\mathcal{T})}^2 \leq \sigma. \tag{6.31}$$

From here, notice that $A \leq \|u * g_0\|_{L^2}^2 + \|\{u * g_\lambda\}\|_{l^2 L^2(\Lambda \times \mathbb{R}^d)}^2 \leq \|u * g_0\|_{L^2}^2 + \|\{u * g_\lambda\}\|_{l^2 L^2(\mathcal{S})}^2 + \|\{u * g_\lambda\}\|_{l^2 L^2(\mathcal{T})}^2 \leq \sigma + \|\{u * g_\lambda\}\|_{l^2 L^2(\mathcal{T})}^2$. Doing a similar calculation for v, we see that

$$\|\{u * g_\lambda\}\|_{l^2 L^2(\mathcal{T})}^2, \|\{v * g_\lambda\}\|_{l^2 L^2(\mathcal{S})}^2 \geq A - \sigma. \tag{6.32}$$

Let $f = u + v$ and $g = u - v$. By Lemma 6.3.1, the reverse triangle inequality, and the fact that $(a+b)^2 \leq 2a^2 + 2b^2$ for all $a, b \geq 0$, we have that

$$\begin{aligned}
\|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2 L^2}^2 &\leq \|(f-g) * g_0\|_{L^2}^2 + \|\{|f * g_\lambda| - |g * g_\lambda|\}_\lambda\|_{l^2 L^2(\Lambda \times \mathbb{R}^d)}^2 \\
&= 2\|v * g_0\|_{L^2}^2 + 2\|\{|f * g_\lambda| - |g * g_\lambda|\}_\lambda\|_{l^2 L^2(\mathcal{S})}^2 + 2\|\{|f * g_\lambda| - |g * g_\lambda|\}_\lambda\|_{l^2 L^2(\mathcal{T})}^2 \\
&\leq 4\|v * g_0\|_{L^2}^2 + 4\|\{u * g_\lambda\}_\lambda\|_{l^2 L^2(S)}^2 + 4\|\{v * g_\lambda\}_\lambda\|_{l^2 L^2(T)}^2
\end{aligned}$$

Applying Equations 6.31 and the bi-Lipschitz bound, we thus have that

$$\gamma^2 \|f - g\|_{L^2/\sim}^2 \leq \|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2 L^2}^2 \leq 8\sigma.$$

Now, we shall bound $\|f - g\|_{L^2/\sim}$ below. Fix $\nu \in \mathbb{C}$, $|\nu| = 1$, and by applying the frame property, we have that $\|f - \nu g\|_{L^2}^2 \geq \|(f - \nu g) * g_0\|_{L^2}^2 + \|\{(f - \nu g) * g_\lambda\}_\lambda\|_{l^2 L^2(\Lambda \times \mathbb{R}^d)}^2 \geq \frac{1}{2}(\|(f - \nu g) * g_0\|_{L^2}^2 + \|\{(f - \nu g) * g_\lambda\}_\lambda\|_{l^2 L^2(\mathcal{S})}^2) + \frac{1}{2}(\|(f - \nu g) * g_0\|_{L^2}^2) + \|\{(f - \nu g) * g_\lambda\}_\lambda\|_{l^2 L^2(\mathcal{T})}^2)$. Using

the reverse triangle inequality, we thus have

$$\begin{aligned}\|f-\nu g\|_{L^2}^2 \geq & \frac{|1+\nu|^2}{2}\|\{v*g_\lambda\}_\lambda\|_{l^2L^2(\mathcal{S})}^2-\frac{|1-\nu|^2}{2}\|\{u*g_\lambda\}_\lambda\|_{l^2L^2(\mathcal{S})}^2\\ &+\frac{|1-\nu|^2}{2}\|\{u*g_\lambda\}_\lambda\|_{l^2L^2(\mathcal{T})}^2-\frac{|1+\nu|^2}{2}\|\{v*g_\lambda\}_\lambda\|_{l^2L^2(\mathcal{T})}^2\\ &-\frac{|1-\nu|^2}{2}\|u*g_0\|_{L^2}^2-\frac{|1+\nu|^2}{2}\|v*g_0\|_{L^2}^2.\end{aligned}$$

Applying Equations 6.31 and 6.32, we thus have

$$\|f-\nu g\|_{L^2}^2 \geq \frac{|1+\nu|^2+|1-\nu|^2}{2}(A-2\sigma)=2(A-2\sigma).$$

Since we assumed $\sigma \leq \frac{A}{4}$, this term is positive. and bounded below by A. Since ν was arbitrary, we thus have that $\|f-g\|_{L^2/\sim}^2 \geq A$, and so

$$\gamma^2 \leq \sigma\frac{8}{A}.$$

Since $\sigma \in (\sigma_{\mathcal{G}}, \frac{A}{4}]$ was arbitrary, we thus have that

$$\gamma^2 \leq \sigma_{\mathcal{G}}\frac{8}{A},$$

as claimed. □

This theorem shows that it is necessary that $\mathcal{G}$ satisfies the σ-SCP for some $\sigma > 0$ in order for $S[\mathcal{P}]$ to have a global lower bi-Lipschitz bound. However, as in the case of phase retrieval, we can show that there exists no such σ by showing that $\sigma_{\mathcal{G}} = 0$ (cf. Theorem 3.13 in [5]).

Theorem 6.3.6. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda\in\Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a semi-discrete frame with lower frame bound $A > 0$ and upper frame bound 1. Then, $\sigma_{\mathcal{G}} = 0$.*

Proof. Let $\eta > 0$ be arbitrary, and let $f \in L^2(\mathbb{R}^d)$ such that $\|f\|_{L^2} = 1$ and $\|f * g_0\|_{L^2}^2 \leq \frac{\eta}{2}$ (such an f exists because $\widehat{g_0}(\xi) \to 0$ as $|\xi| \to \infty$). By the frame property, we have that

$$\|f * g_0\|_{L^2}^2 + \sum_{\lambda \in \Lambda} \|f * g_\lambda\|_{L^2}^2 \leq \|f\|_{L^2}^2 < \infty.$$

So, since Λ is countable, we can order the elements and easily conclude that there must exist a finite subset $\Lambda' \subseteq \Lambda$ such that

$$\sum_{\lambda \in \Lambda \setminus \Lambda'} \|f * g_\lambda\|_{L^2}^2 \leq \frac{\eta}{2}.$$

Take $\mathcal{S} = \Lambda' \times \mathbb{R}^d$. Then, the previous inequalities imply that

$$\|f * g_0\|_{L^2}^2 + \|\{f * g_\lambda\}_\lambda\|_{l^2 L^2(\mathcal{T})}^2 < \eta, \tag{6.33}$$

where $\mathcal{T} = (\Lambda \times \mathbb{R}^d) \setminus \mathcal{S}$. Now, we have finitely many remaining filters $\{g_0\} \cup \{g_{\lambda'}\}_{\lambda' \in \Lambda'}$, all of whose Fourier transforms converge to 0 as $|\xi| \to \infty$ by the Riemann-Lebesgue lemma. In particular, $|\widehat{g_0}(\xi)|^2 + \sum_{\lambda' \in \Lambda'} |\widehat{g_{\lambda'}}(\xi)|^2 \to 0$ as $|\xi| \to \infty$. Thus, using the Plancherel Theorem, there must exist some $g \in L^2(\mathbb{R}^d)$, $\|g\|_{L^2} = 1$ such that

$$\|g * g_0\|_{L^2}^2 + \|\{g * g_\lambda\}_\lambda\|_{l^2 L^2(\mathcal{S})}^2 < \eta. \tag{6.34}$$

Thus, we have $A_{\mathcal{S}}, A_{\mathcal{T}} \leq \eta$. Since $\eta > 0$ was arbitrary, we must have that $\sigma_{\mathcal{G}} = 0$. □

Combining Theorems 6.3.5 and 6.3.6, we see that $\gamma = 0$ and so no global lower bi-Lipschitz bound can exist.

6.3.2 Subspace and Local Bi-Lipschitz Bounds

While no global lower bi-Lipschitz bounds may exist with respect to the $L^2/\sim$ norm, it's still possible that bounds could exist on subspaces of $L^2(\mathbb{R}^d)$ or locally. In the next proposition, we present a class of examples of subspaces for which bi-Lipschitz bounds do exist for some constructions of scattering transforms.

Proposition 6.3.7. *Let $\mathcal{G} = \{g_0\} \cup \{g_\lambda\}_{\lambda\in\Lambda} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ be a semi-discrete Bessel sequence with upper frame bound 1. Suppose that there exists some $\gamma \in (0,1)$ and $R > 0$ such that $|\widehat{g_0}(\xi)| \geq \gamma$ for all $\xi \in Q_R(0)$. Then, for all R-bandlimited functions $f, g \in L^2(\mathbb{R}^d)$, i.e., $\text{supp}(\widehat{f}), \text{supp}(\widehat{g}) \subseteq Q_R(0)$, we have that $S[\mathcal{P}]$ is bi-Lipschitz with*

$$\gamma\|f-g\|_{L^2} \leq \|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2L^2}. \tag{6.35}$$

Proof. Notice by definition that

$$\|S[\mathcal{P}]f - S[\mathcal{P}]g\|_{l^2L^2}^2 \geq \|f*g_0 - g*g_0\|_{L^2}^2.$$

Using the Plancherel Theorem and the fact that f, g are R-bandlimited, we see that

$$\|f*g_0 - g*g_0\|_{L^2}^2 = \int\limits_{Q_R(0)} |\widehat{f}(\xi)-\widehat{g}(\xi)|^2|\widehat{g_0}(\xi)|^2 \mathrm{d}\xi \geq \gamma^2 \int\limits_{Q_R(0)} |\widehat{f}(\xi)-\widehat{g}(\xi)|^2 \mathrm{d}\xi = \gamma^2\|f-g\|_{L^2}^2.$$

□

In particular, we see that the GFSTs generated by frequency shifts of Gaussians (see Exam-

ple 6.2.3) are bi-Lipschitz for band-limited functions, though the lower b-Lipschitz constant decays exponentially fast as the bandlimiting parameter $R \to \infty$.

Thus, we have constructed invertible scattering transforms whose inverse is stable on particular subspaces of $L^2(\mathbb{R}^d)$. However, there remain open questions as to whether one can construct invertible scattering transforms with g_0 bandlimited, or whether local lower bi-Lipschitz bounds can exist for the inverse. We are particularly interested in these questions in the case that the scattering transforms considered are FSTs or GFSTs.

Once again, the field of (frame) phase retrieval may hold the key to answering these questions. Many papers have been dedicated to the question of when functions can be recovered from the modulus of their frame coefficients [58, 90], and in particular when the frame is a Gabor frame [7, 59, 120]. The limits of stability have also been considered in the latter case, most notably when the frames are generated by time-frequency shifts of Gaussians [3, 4, 6, 60]. Given that we were able to extend the results of [5] to our setting, it's possible that we will likewise be able to use similar arguments as in the above papers to explore the possibilities and limits of stably inverting Fourier and generalized Fourier scattering transforms.

6.4 Numerical Inversions of GFSTs

While we have so far studied theoretical aspects of GFSTs, we are also interested in how well they perform in applications. Thus, we study two different GFSTs and methods for inverting them. In Section 6.4.1, we study whether a similar DCGAN architecture as used to inver the discretized WST can invert a discretized FST (which is, of course, an example of a discretized GFST). In Section 6.4.2, we experiment with inverting a discretized GFST generated by frequency shifts of Gaussians using the

process described in the proof of Theorem 6.2.9.

6.4.1 Inverting the Discretized FST with a DCGAN

In our first attempts at inverting a generalized Fourier scattering transform (actually an FST), we consider a similar process that was used for inverting the discretized WST. As in Section 5.2.3 and [14], we explore how a DCGAN network may be used to invert a discretized FST. In this section, we provide details on the discretized FST architecture that we use, and we compare the performance of the proposed inverse of the FST to the same inverse for the WST on the ARAD dataset.

6.4.1.1 A Discretized FST

For our discretized FST, we use a similar architecture as proposed in [70, 71]. This proposed FST was built for 3-dimensional data, and it was implemented using TensorFlow. In order to make the transform easily work with our setting, we recoded it using PyTorch and for 2-dimensional data, e.g., images.

Let $M_h, N_w \in \mathbb{N}$ be chosen (odd number) hyperparameters that control the number of frequency shifts that we calculate in the height and width dimensions, respectively. Then, we define $g_0 = g_{M_h,N_w}$ to be the $M_h \times N_w$ convolutional window which identically equal to $\frac{1}{M_h M_w}$. That is, $g_0(m,n) = \frac{1}{M_h M_w}$ for $1 \leq m \leq M_h$, $1 \leq n \leq N_w$. For $1 \leq \lambda_h \leq M_h, 1 \leq \lambda_w \leq N_w$, we define the additional filters

$$g_{(\lambda_h,\lambda_w)}(m,n) = \exp\left(2\pi i \left(\frac{\lambda_h m}{M_h} + \frac{\lambda_w n}{N_w}\right)\right) g_0(m,n). \tag{6.36}$$

These are the same convolutional filters used in [70, 71], with two exceptions. First, our filters are designed for 2-dimensional data (i.e., images), while theirs were designed for 3-dimensional data.

Second, their filters were normalized so that they had constant l^2 norm, while we normalize ours so that it has constant l^1 norm. We make this choice both so that it lines up with the choice to normalize the l^1 norm of the convolutional filters for the discretized WST (Section 2.4.1) and also so that if a (rescaled) image F is in the range $[-1, 1]$, then $F * g_{(\lambda_h, \lambda_w)}$ is still in the range $[-1, 1]$ by Young's convolutional inequality.

Note that if F is an $M \times N$ image (and so real-valued), then $|F \star g_{(\lambda_h, \lambda_w)}| = |F \star g_{(M_h - \lambda_h, N_w - \lambda_w)}|$. This is because $\overline{F} = F, \overline{g_0} = g_0$, and $\overline{\exp\left(2\pi i \left(\frac{\lambda_h m}{M_h} + \frac{\lambda_w n}{N_w}\right)\right)} = \exp\left(2\pi i \left(\frac{-\lambda_h m}{M_h} + \frac{-\lambda_w n}{N_w}\right)\right) = \exp\left(2\pi i \left(\frac{(M_h - \lambda_h) m}{M_h} + \frac{(N_w - \lambda_w) n}{N_w}\right)\right)$. Thus, we define the discretized FST of F, $S[M_h, N_w]F$, as the union of the following three sets:

$$\{F \star g_0\} \tag{6.37}$$

$$\{|F \star g_{(\lambda_h, \lambda_w)}| \star g_0\}_{\substack{1 \leq \lambda_h \leq M_h/2 \\ 1 \leq \lambda_w \leq N_w}} \tag{6.38}$$

$$\{||F \star g_{(\lambda_h, \lambda_w)}| \star g_{(\lambda'_h, \lambda'_w)}| \star g_0\}_{\substack{1 \leq \lambda_h \leq \lambda'_h \leq M_h/2 \\ 1 \leq \lambda_w \leq \lambda'_w \leq N_h \\ \lambda_h < \lambda'_h \text{ or } \lambda_w < \lambda'_w}} \tag{6.39}$$

Note that, as in [70, 71], we only calculate the second-layer coefficients if both $\lambda_h \leq \lambda'_h, \lambda_w \leq \lambda'_w$ and at least one of these inequalities is strict. Additionally, we only calculate the outputs for $\lambda_h, \lambda'_h \leq M_h/2$ as the remaining terms are equal to one of the already calculated terms by the discussion above.

As in the case of the discretized WST, the discretized FST is a highly redundant representation. Thus, we downsample each spatial dimension by a factor of $2^2 = 4$; we choose to downsample twice because we are calculating 2 layers of scattering coefficients. Thus, if the image is of size $\widehat{C} \times M \times N$, where $\widehat{C}$ is the number of channels and M, N are the spatial dimensions, then the output of the discretized FST will be of size $\left(\widehat{C}, S, \frac{M}{4}, \frac{N}{4}\right)$, where S is the number of Fourier scattering channels.

6.4.1.2 Inversion Network Architecture

In order to invert this discretized FST, we explore the use of the same DCGAN architecture used to invert the discretized WST. As described in Section 5.2.3.2, this network consists of several convolutional blocks and upsamplings. As we downsample our FST output by the same factor as used for the discretized WST in this section, we also use two convolutional blocks consisting of upsampling, a convolutional layer (without bias), batch normalization, and ReLU activation, followed by one more block consisting of a convolutional layer (without bias), batch normalization, and $\tanh$ activation. See Figure 5.6 for a diagram of this network, which is the same architecture used for inverting the discretized FST (though with different numbers of convolutional filters).

For the filters $g_0, g_{(\lambda_1,\lambda_2)}$, we use a window with height and width $M_h = N_w = 5$. In that case, there are $S = 31$ scattering channels. We train the inverse network for 1000 epochs with a batch size of 2. We use the AdamW optimizer with weight decay parameter 10^{-5}, and we consider two different initial learning rates, 10^{-3} and 10^{-4}. Finally, we consider both l^1 and l^1 + CD loss functions.

6.4.1.3 Experiments and Results

Due to computational restraints, we were only able to test the proposed inverse network for the discretized FST on the ARAD dataset. As in the case of hyperspectral reconstruction, we trained the inverse network to invert the FST embedding of the hyperspectral images. We trained 4 different versions of the inverse network, depending on the combination of initial learning rate and loss function as described in the Section 6.4.1.2.

Our results from these experiments are in Table 6.1; here, ST refers to scattering transform. We compare the 4 different FST inverse networks with their corresponding WST networks, and we report

Inverse			Validation Set		Testing Set	
ST	**Loss**	**LR**	**Average SAM**	**Average SSIM**	**Average SAM**	**Average SSIM**
WST	0 CD	10^{-4}	0.0923 ± 0.0674	0.9541 ± 0.0339	0.1044 ± 0.0641	0.9418 ± 0.0480
WST	0 CD	10^{-3}	0.1478 ± 0.1623	0.9637 ± 0.0352	0.1878 ± 0.1845	0.9580 ± 0.0537
WST	1 CD	10^{-4}	0.0753 ± 0.0506	0.9459 ± 0.0461	0.0925 ± 0.0651	0.9189 ± 0.0673
WST	1 CD	10^{-3}	0.0702 ± 0.0458	0.9596 ± 0.0448	0.0845 ± 0.0554	0.9458 ± 0.0602
FST	0 CD	10^{-4}	0.1081 ± 0.0883	0.9415 ± 0.0806	0.1418 ± 0.1126	0.9210 ± 0.1032
FST	0 CD	10^{-3}	0.2061 ± 0.2038	0.9630 ± 0.0402	0.2536 ± 0.2093	0.9427 ± 0.0881
FST	1 CD	10^{-4}	0.0811 ± 0.0480	0.9403 ± 0.0573	0.0993 ± 0.0682	0.9158 ± 0.0742
FST	1 CD	10^{-3}	0.0707 ± 0.0476	0.9346 ± 0.1114	0.0875 ± 0.0690	0.9259 ± 0.1050

Table 6.1: Results from our experiments on inverting the discretized FST with a DCGAN network on ARAD. We compare the inversion results for the FST and WST. All networks were trained for 1000 epochs with a batch size of 2 and weight decay parameter 10^{-5}

the results for both our validation and testing sets for ARAD. The first thing to note is that the inverse networks are performing noticeably better when inverting the discretized WST than when inverting the discretized FST on both the validation and testing sets. This suggests that the DCGAN network is not optimal for inverting the discretized FST, and that a different method is needed. It should also be noted that the SSIM results when inverting the WST or the FST are significantly better than the SSIM results of the full Hyperscat pipeline (see Table 5.33).

6.4.2 Inverting the Discretized GFST Using the DFT

In Section 6.2, we constructed generalized Fourier scattering transforms with an easily calculated inverse. In this section, we examine how the same transforms and inverse process perform numerically. In particular, we consider discretized GFSTs which use frequency shifts of Gaussians as the convolutional filters.

6.4.2.1 A Discretized GFST

Our construction of a discretized GFST will be similar to the previously discussed construction of a discretized FST (see Section 6.4.1.1). We start with our output-generating function g_0, which is constructed using a discretized version of a Gaussian $e^{-\pi^2|x|^2}$. However, unlike the discretized FST (but like the discretized WST), the size of this convolutional window is equal to the size of the input image (which is specified as a hyperparameter). This is done in order to make it easier to calculate the discrete Fourier transform (DFT) of g_0 during the inversion process. Like the discretized FST, we normalize g_0 so that it has an l^1 norm of 1.

For the other filters, two hyperparameters are chosen, $\mathcal{M}_h, \mathcal{N}_w \in \mathbb{N}$, which control the number of frequency shifts that will be considered. For $1 \leq \lambda_h \leq \mathcal{M}_h, 1 \leq \lambda_w \leq \mathcal{N}_w$, the filters $g_{(\lambda_h,\lambda_w)}$ are constructed from the frequency shifts of g_0, i.e., from $\exp\left(2\pi i \left(\frac{\lambda_h m}{\mathcal{M}_h} + \frac{\lambda_w n}{\mathcal{N}_w}\right)\right) g_0(m,n)$. These are also normalized to have l^1 norm 1. Then, for a single 2-dimensional image F of size $M \times N$, its discretized GFST $S[\mathcal{M}_h, \mathcal{N}_w]F$ is given by the union of the following three sets:

$$\{F \star g_0\} \tag{6.40}$$

$$\{|F \star g_{(\lambda_h,\lambda_w)}| \star g_0\}_{\substack{1 \leq \lambda_h \leq \mathcal{M}_h/2 \\ 1 \leq \lambda_w \leq \mathcal{N}_w}} \tag{6.41}$$

$$\{||F \star g_{(\lambda_h,\lambda_w)}| \star g_{(\lambda'_h,\lambda'_w)}| \star g_0\}_{\substack{1 \leq \lambda_h \leq \lambda'_h \leq \mathcal{M}_h/2, \\ 1 \leq \lambda_w \leq \lambda'_w \leq \mathcal{N}_h \\ \lambda_h < \lambda'_h \text{ or } \lambda_w < \lambda'_w}} \tag{6.42}$$

Unlike the discretized WST or FST, however, we choose to not downsample the generalized GFST due to our use of the DFT during the inversion process. Thus, if F is an image of size $\widehat{C} \times M \times N$, then $S[\mathcal{M}_h, \mathcal{N}_w]F$ is of size $(\widehat{C}, S, M, N)$, where S is the number of GFST scattering channels.

Validation Set		Testing Set	
Average SAM	**Average SSIM**	**Average SAM**	**Average SSIM**
0.0001 ± 0.0000	1.0000 ± 0.0000	0.0001 ± 0.0000	1.0000 ± 0.0000

Table 6.2: Results from our experiments on inverting the discretized GFST using the DFT. This method requires no training.

6.4.2.2 Inverting the Discretized GFST: Method and Numerical Results

To invert the discretized GFST, we employ the same process that is used to invert the GFST in the continuous setting in Theorem 6.2.9. Namely, given $S[\mathcal{M}_h, \mathcal{N}_w]F$, we look at its 0-th layer coefficients $F \star g_0$, which is of size $(\widehat{C}, M, N)$. We take the DFT of each channel F_c for $1 \leq c \leq \widehat{C}$ of this term to get $\widehat{F}_c \, \widehat{g}_0$. Since $\widehat{g}_0$ is non-zero everywhere, we can divide by it to get $\widehat{F}_c$. Taking the inverse discrete Fourier transform (IDFT) reconstructs F_c. Since we do this for each c, we thus have reconstructed the entire image F. In our experiments, we use $\mathcal{M}_h = \mathcal{N}_w = 4$, so $S = 19$.

We test the accuracy of our inverse algorithm on the HSI of the ARAD dataset; our results for both the validation and testing sets are in Table 6.2. As our new algorithm amounts to inverting the DFT (which is well known to be invertible, with an easily calculable inverse), one would expect the inverse of the GFST to have high accuracy. Indeed, we see that the inverse reconstructs the images nearly perfectly, and we expect that any errors that have arisen can be attributed to floating point arithmetic.

6.4.3 Conclusion and Future Work

In this chapter, I constructed generalized Fourier scattering transforms and proved that they have many of the same properties as Fourier scattering transforms, including that their K-th layer energy decays exponentially fast in K. I also showed that certain constructions of GFSTs are invertible in the

continuous setting, though not stably. Finally, I demonstrated numerically that the inverse of GFSTs is nearly perfect, as opposed to our initial attempts of inverting a discretized FST.

Part of our future work will be dedicated to finding an algorithm for inverting the discretized FST. We have shown in Section 6.4.1 that the DCGAN algorithm proposed by Mallat and Angle for inverting the discretized WST is not sufficient for inverting the discretized FST. It is possible that ideas from phase retrieval could be useful here in finding an algorithm for inverting the discretized FST, with the added benefit that such an inverse might not need to be trained.

Beyond the FST inverse, we also plan to experiment with replacing the WST and its inverse in our current Hyperscat pipeline with the discretized FST or GFST (and their corresponding inverses). FSTs have previously been shown to perform notably better on hyperspectral classification tasks than WSTs or various neural network-based methods [70, 71], and so we conjecture that FSTs or GFSTs could similarly be used to improve hyperspectral reconstruction performance when compared with WSTs. At the same time, while the inversion algorithm for our discretized GFST is nearly perfect, we expect that there may be issues when training our current matching networks to match GFST coefficients due to their larger spatial size. Thus, we will also explore more complex and powerful neural network structures, such as transformers, for improving our matching results.

Bibliography

[1] J. Aeschbacher, J. Wu, and R. Timofte. In defense of shallow learned spectral reconstruction from RGB images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 471 – 479, 2017.

[2] N. Akhtar and A. Mian. Hyperspectral recovery from RGB images using Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42:100–113, 2018.

[3] R. Alaifari, F. Bartolucci, S. Steinerberger, and M. Wellershoff. On the connection between uniqueness from samples and stability in Gabor phase retrieval. Sampling Theory, Signal Processing, and Data Analysis, 22(6), 2024.

[4] R. Alaifari, I. Daubechies, P. Grohs, and R. Yin. Stable phase retrieval in infinite dimensions. Foundations of Computational Mathematics, 19:869–900, 2018.

[5] R. Alaifari and P. Grohs. Phase retrieval in the general setting of continuous frames for Banach spaces. SIAM Journal on Mathematical Analysis, 49(3):1895–1911, 2017.

[6] R. Alaifari, B. Pineau, M. A. Taylor, and M. Wellershoff. Cheeger's constant for the Gabor transform and ripples, 2025. arXiv:2512.18058.

[7] R. Alaifari and M. Wellershoff. Uniqueness of STFT phase retrieval for bandlimited functions. Applied and Computational Harmonic Analysis, 50:34–48, 2021.

[8] L. Alzubaidi et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(53), 2021.

[9] J. Andén, V. Lostanlen, and S. Mallat. Joint time-frequency scattering for audio classification. In IEEE International Workshop on Machine Learning for Signal Processing (MLSPW), 2015.

[10] J. Andén, V. Lostanlen, and S. Mallat. Classification with joint time-frequency scattering. IEEE Transactions on Signal Processing, 17(4), 2019.

[11] J. Andén and S. Mallat. Scattering representations of modulated sounds. In 15th International Conference on Digital Audio Effects (DAFx-12), 2012.

[12] J. Andén and S. Mallat. Deep scattering spectrum. IEEE Transactions on Signal Processing, 62(16):4114–4128, 2014.

[13] M. Andreux et al. Kymatio: Scattering transforms in Python. Journal of Machine Learning Research, 21(60):1–6, 2020.

[14] T. Angles and S. Mallat. Generative networks as inverse problems with scattering transforms. In Proceedings of the International Conference on Learning Representations, 2018.

[15] B. Arad and O. Ben-Shahar. Sparse recovery of hyperspectral signal from natural RGB images. In European Conference on Computer Vision, pages 19 – 34, 2016.

[16] B. Arad et al. Ntire 2022 spectral recovery challenge and data set. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 863–881, June 2022.

[17] Z. Baharlouei, H. Rabbani, and G. Plonka. Wavelet scattering transform application in classification of retinal abnormalities using OCT images. Scientific Reports, 13(19013), 2023.

[18] R. Balan, M. Singh, and D. Zou. Lipschitz properties for deep convolutional networks. In Frames and Harmonic Analysis: AMS Special Sessions on Frames, Wavelets, and Gabor Systems and Frames, Harmonic Analysis, and Operator Theory, April 16–17, 2016, North Dakota State University, Fargo, North Dakota, number 706 in Contemporary Mathematics, pages 129–151, Providence, RI, 2018. American Mathematical Society.

[19] J. J. Benedetto. Harmonic Analysis and Applications. CRC Press, 1996.

[20] J. J. Benedetto and W. Czaja. Integration and Modern Analysis. Birkhäuser, 2009.

[21] A. Bhargava, A. Sachdeva, K. Sharma, M. H. Alsharif, P. Uthansakul, and M. Uthansakul. Hyperspectral imaging and its applications: A review. Heliyon, 10(10), 2024.

[22] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. In Proceedings of the 35th International Conference on Machine Learning (ICML), 2018.

[23] P. J. Brockwell and R. A. Davis. Time Series: Theory and Methods. Springer-Verlag, 1987.

[24] J. Bruna. Scattering Representations for Recognition. PhD thesis, École Polytechnique, 2012.

[25] J. Bruna. The Scattering Transform, page 338–399. Cambridge University Press, 2022.

[26] J. Bruna and S. Mallat. Invariant scattering convolution networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1872–1886, 2013.

[27] W. G. Buratto et al. Wavelet CNN-LSTM time series forecasting of electricity power generation considering biomass thermal systems. IET Generation, Transmission, and Distribution, 2024.

[28] Y. Cai et al. Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17481 – 17490, 2022.

[29] Y. Cai et al. MST++: Multi-stage spectral-wise transformer for efficient spectral reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 745–755, 2022.

[30] B. Chen, K. Miller, A. L. Bertozzi, and J. Schwenk. CGAP: A hybrid contrastive and graph-based active learning pipeline to detect water and sediment in multispectral images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18:446–462, 2025.

[31] X. Chen, X. Cheng, and S. Mallat. Unsupervised deep Haar scattering on graphs. In Conference on Neural Information Processing Systems (NeurIPS), 2014.

[32] A. Chua. Towards time-frequency deformation stability bounds for deep convolutional neural networks. In ICASSP 2025 - ICASSP 2025 IEEE International Conference on Acoustics, Speech and Signal Processing, 2025.

[33] A. Cloninger. Exploiting Data-Dependent Structure for Improving Sensor Acquisition and Integration. PhD thesis, University of Maryland - College Park, 2014.

[34] A. Cloninger, W. Czaja, and T. Doster. A case study on data fusion with overlapping segments. In Applied Imagery Pattern Recognition Workshop, 2013.

[35] A. Cloninger, W. Czaja, and T. Doster. Operator analysis and diffusion based embeddings for heterogeneous data fusion. In IEEE International Symposium on Geoscience and Remote Sensing, 2014.

[36] A. Cloninger, W. Czaja, and T. Doster. Multimodal data fusion via graph- and operator-based techniques. Hyperspectral Imaging and Sounding of the Environment, 2015.

[37] A. Cloninger, W. Czaja, and T. Doster. The pre-image problem for Laplacian eigenmaps utilizing l_1 regularization with applications to data fusion. Inverse Problems, 33(7):074006, 2017.

[38] R. R. Coifman and M. J. Hirn. Diffusion maps for changing data. Applied and Computational Harmonic Analysis, 17(3):79–107, 2014.

[39] W. Czaja and J. Emidih. Heterogeneous cancer cell line data fusion for identifying novel response determinants in precision medicine. In Bioinformatics Research and Applications, pages 414–419. Springer International Publishing, 2017.

[40] W. Czaja, J. Emidih, B. Kolstoe, and R. G. Spencer. Hyperspectral reconstruction of skin through fusion of scattering transform features. In ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024.

[41] W. Czaja, J. Emidih, B. Kolstoe, and R. G. Spencer. Data fusion of scattering transform features for hyperspectral image reconstruction. Open Journal of Signal Processing, Submitted.

[42] W. Czaja, B. Kolstoe, and D. Koralov. Stationary processes scattering. SIAM Journal on Mathematical Analysis, Submitted.

[43] W. Czaja, B. Kolstoe, and D. Koralov. Translation-invariant behavior of general scattering transforms. Comptes Rendus Mathématique, Submitted.

[44] W. Czaja and W. Li. Analysis of time-frequency scattering transforms. Applied and Computational Harmonic Analysis, 47(1):149–171, 2019.

[45] W. Czaja and W. Li. Rotationally invariant time-frequency scattering transforms. Journal of Fourier Analysis and Applications, 26(4), 2020.

[46] H. Deborah, N. Richard, and H. Y. Hardeberg. A comprehensive evaluation of spectral distance functions and metrics for hyperspectral image processing. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(6):3224 – 3234, 2015.

[47] R. Dian, T. Shan, W. He, and H. Liu. Spectral super-resolution via model-guided cross-fusion network. IEEE Transactions on Neural Networks and Learning Systems, 35(7):10059 – 10070, 2024.

[48] J. L. Doob. Stochastic Processes. John Wiley & Sons, 1953.

[49] T. Doster. Harmonic Analysis Inspired Data Fusion for Applications in Remote Sensing. PhD thesis, University of Maryland - College Park, 2014.

[50] M. Eickenberg, G. Exarchakis, M. Hirn, and S. Mallat. Solid harmonic wavelet scattering: Predicting quantum molecular energy from invariant descriptors of 3D electronic densities. In Conference on Neural Information Processing Systems (NeurIPS), 2017.

[51] J. Emidih. Graph-Based Data Fusion with Applications to Magnetic Resonance Imaging. PhD thesis, University of Maryland - College Park, 2023.

[52] H. Führ and M. Getter. Energy propagation in scattering convolution networks can be arbitrarily slow. Applied and Computational Harmonic Analysis, 79, 2025.

[53] F. Gao, M. Hirn, M. Perlmutter, and G. Wolf. Geometric wavelet scattering on graphs and manifolds. In Proceedings of SPIE, Wavelets and Sparsity XVIII, 2019.

[54] F. Gao, G. Wolf, and M. Hirn. Geometric scattering for graph data analysis. In Proceedings of Machine Learning Research (PMLR), 2019.

[55] A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney, and K. Keutzer. AI and memory wall. IEEE Micro, 44(3):33–39, 2024.

[56] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016. `http://www.deeplearningbook.org`.

[57] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 2672–2680. MIT Press, 2014.

[58] P. Grohs, S. Koppensteiner, and M. Rathmair. Phase retrieval: Uniqueness and stability. SIAM Review, 62(2):301–350, 2020.

[59] P. Grohs and L. Liehr. Injectivity of Gabor phase retrieval from lattice measurements. Applied and Computational Harmonic Analysis, 62:173–193, 2023.

[60] P. Grohs and M. Rathmair. Stable gabor phase retrieval for multivariate functions. Journal of the European Mathematical Society, (5):1593–1615, 2021.

[61] D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs), 2023. arXiv:1606.08415.

[62] S. Heydari and Q. H. Mahmoud. Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions. Sensors, 25(10), 2025.

[63] M. Hirn, N. Poilvert, and S. Mallat. Quantum energy regression using scattering transforms, 2016. arXiv:1502.02077.

[64] D. Hong, C. Li, N. Yokoya, B. Zhang, X. Jia, A. Plaza, P. Gamba, J. A. Benediktsson, and J. Chanussot. Hyperspectral imaging, 2025. arXiv:2508.08107.

[65] H. Hotelling. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24(6):417–441, 1933.

[66] X. Hu et al. Hdnet: High-resolution dual-domain learning for spectral compressive imaging. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17521 – 17530, 2022.

[67] Y. Jia et al. From RGB to spectrum for natural scenes via manifold-based mapping. In Proceedings of the IEEE International Conference on Computer Vision, pages 4705–4713, 2017.

[68] Z. Jiang, W. Zhang, and W. Wang. Fusiform multi-scale pixel self-attention network for hyperspectral images reconstruction from a single RGB image. The Visual Computer, 39:3573–3584, 2023.

[69] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, and J. Kong. Prediction for time series with CNN and LSTM. In Proceedings of the 11th International Conference on Modeling, Identifiction, and Control (ICMIC2019), pages 631–641, 2019.

[70] I. Kavalerov. Impact of Semantic Physics, and Adversarial Mechamisms in Deep Learning. PhD thesis, University of Maryland - College Park, 2020.

[71] I. Kavalerov, W. Li, W. Czaja, and R. Chellappa. 3-D Fourier scattering transform and classification of hyperspectral images. Transactions on Geoscience and Remote Sensing, 59(12):10312–10327, 2021.

[72] B. Kaya, Y. B. Can, and R. Timofte. Towards spectral estimation from a single RGB image in the wild. In IEEE International Conference on Computer Vision, pages 3546–3555, 2019.

[73] M. J. Khan et al. Modern trends in hyperspectral image analysis: A review. IEEE Access, 6:14118 – 14129, 2018.

[74] L. B. Koralov and Y. G. Sinai. Theory of Probability and Random Processes. Springer, 2nd edition, 2007.

[75] B. Lavrič. Continuity of monotone functions. Archivum Mathematicum, 29(1-2):1–4, 1993.

[76] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521:436–444, 2015.

[77] J. Li, C. Wu, R. Song, Y. Li, and F. Liu. Adaptive weighted attention network with camera spectral sensitivity prior for spectral reconstruction from RGB images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 1894–1903, 2020.

[78] M. Li, Y. Fu, J. Liu, and Y. Zhang. Pixel adaptive deep unfolding transformer for hyperspectral image reconstruction. In IEEE International Conference on Computer Vision, 2023.

[79] W. Li. Topics in Harmonic Analysis, Sparse Representations, and Data Analysis. PhD thesis, University of Maryland - College Park, 2018.

[80] Y. Li. Feature Extraction in Image Processing and Deep Learning. PhD thesis, University of Maryland - College Park, 2018.

[81] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 1132–1140, 2017.

[82] Y. Liu, J. Zhang, and Y. Zhang. Hyperspectral reconstruction from a single textile RGB image based on the generative adversarial network. Textile Research Journal, 93(1-2):307–316, 2023.

[83] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vission (ICCV), 2015.

[84] I. E. Livieris, E. Pintelas, and P. Pintelas. A CNN-LTSM model for gold price time-series forecasting. Neural Computing and Applications, 32:17351–17360, 2020.

[85] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations (ICLR), 2019.

[86] V. Lostanlen. Convolutional Operaters in the Time-Frequency Domain. PhD thesis, École normale supérieure, 2017.

[87] M. Magnusson, J. Sigurdsson, S. E. Armansson, M. O. Ulfarsson, H. Deborah, and J. R. Sveinsson. Creating RGB images from hyperspectral images using a color matching function. In IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, pages 2045–2048, 2020.

[88] S. Mallat. Group invariant scattering. Communications in Pure and Applied Mathematics, 65(10):1331–1398, 2012.

[89] S. Mallat. Understanding deep convolutional networks. Philosophical Transactions of the Royal Society A, 374(2065), 2016.

[90] S. Mallat and I. Waldspurger. Phase retrieval for the Cauchy wavelet transform. Journal of Fourier Analysis and Applications, 21:1251–1309, 2015.

[91] J. Muprhy. Anisotropic Harmonic Analysis and Integration of Remotely Sensed Data. PhD thesis, University of Maryland - College Park, 2015.

[92] P. C. Ng et al. Hyper-Skin: A hyperspectral dataset for reconstructing facial skin-spectra from RGB images. In Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023.

[93] P. C. Ng et al. Hyperspectral skin vision challenge: Can your camera see beyond your skin? In ICASSP 2024 SPGC IEEE International Conference on Acoustics, Speech and Signal Processing Grand Challenge, 2024.

[94] F. Nicola and S. I. Trapasso. Stability of the scattering transform for deformations with minimal regularity. Journal de Mathématiques Pures et Appliquées, pages 122–150, 2023.

[95] F. Nicola and S. I. Trapasso. Generalized moduli of continuity under irregular or random deformations via multiscale analysis. Information and Inference: a Journal of the IMA, 14(2), 2025.

[96] E. Oyallon. Analyzing and Introducing Structures in Deep Convolutional Neural Networks. PhD thesis, École normale supérieure - PSL Research University, 2017.

[97] E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In IEEE International Conference on Computer Vision (ICCV), pages 5619–5628, 2017.

[98] E. Oyallon, S. Mallat, and L. Sifre. Generic deep networks with wavelet scattering. In International Conference on Learning Representations Workshop (ICLRW), 2014.

[99] M. Panahi et al. Deep learning neural networks for spatially explicit prediction of flash flood probability. Geoscience Frontiers, 12, 2021.

[100] A. Papoulis and S. U. Pillai. Probability, Random Variables, and Stochastic Processes. McGraw-Hill, 4th edition, 2002.

[101] K. Pearson. On lines and planes of closest fit to systems of points in space. Philosophical Magazine Series 6, 2(11):559–572, 1901.

[102] M. Pekala. Harmonic Analysis and Machine Learning. PhD thesis, University of Maryland - College Park, 2018.

[103] H. Peng, X. Chen, and J. Zhao. Residual pixel attention network for spectral reconstruction from RGB images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 2012–2020, 2020.

[104] M. Perlmutter, F. Gao, G. Wolf, and M. Hirn. Geometric wavelet scattering networks on compact Riemannian manifolds. In Proceedings of Machine Learning Research (PMLR), volume 107, pages 570–604, 2020.

[105] M. Perlmutter, A. Tong, F. Gao, G. Wolf, and M. Hirn. Understanding graph neural networks with generalized geometric scattering transforms. SIAM Journal on Mathematics of Data Science, 5(4), 2023.

[106] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flanner. Numerical recipes in C: the art of scientific computing. Cambridge University Press, 2nd edition, 1992.

[107] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In International Conference on Learning Representations, 2014.

[108] M. K. Reddy and P. Alku. Automatic detection of Parkinsonian speech using wavelet scattering features. JASA Express Letters, 5(055202), 2025.

[109] A. Robles-Kelly. Single image spectral reconstruction for multimedia applications. In Association for Computer Machinery Multimedia, pages 251–260, 2015.

[110] Y. A. Rozanov. Stationary Random Processes. Holden-Day, 1967.

[111] W. Rudin. Fourier Analysis on Groups. Wiley, 1962.

[112] J. Schwenk and J. Rowland. Riverpixels: paired landsat images and expert-labeled sediment and water pixels for a selection of rivers v1.0. Technical report, ESS-DIVE Deep Insight for Earth Science Data, 2021.

[113] Z. Shi et al. Hscnn+: Advanced CNN-based hyperspectral recovery from RGB images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 1052 – 1060, 2018.

[114] L. Sifre and S. Mallat. Combined scattering for rotation invariant texture analysis. In European Symposium on Artificial Neural Networks (ESANN), pages 127–132, 2012.

[115] L. Sifre and S. Mallat. Rotation, scaling, and deformation invariant scattering for texture discrimination. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[116] A. C. C. Silveira, D. S. do Carmo, L. H. Ueda, D. G. Fantinato, P. D. P. Costa, and L. Rittner. VITMST++: Efficient hyperspectral reconstruction through vision transformer-based spatial compression. IEEE Open Journal of Signal Processing, 6:398–404, 2025.

[117] T. Stiebel, S. Koppers, P. Seltsam, and D. Merhof. Reconstructing spectral images from RGB-images using a convolutional neural network. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 1061–1066, 2018.

[118] I. Waldspurger. Exponential decay of scattering coefficients. In 2017 International Conference on Sampling Theory and Applications (SampTA), pages 143–146, 2017.

[119] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[120] M. Wellershoff. Injectivity of sampled Gabor phase retrieval in spaces with general integrability conditions. Journal of Mathematical Analysis and Applications, 530, 2024.

[121] T. Wiatowski and H. Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. IEEE Transactions on Information Theory, 64(3):1845–1866, 2017.

[122] T. Wiatowski, P. Grohs, and H. Bölcskei. Energy propagation in deep convolutional neural networks. IEEE Transactions on Information Theory, 64(7):4819–4842, 2018.

[123] Z. Xiong et al. HSCNN: CNN-based hyperspectral image recovery from spectrally undersampled projections. In IEEE/CVF International Conference on Computer Vision Workshops, pages 518–525, 2017.

[124] Y. Yan, L. Zhang, J. Li, W. Wei, and Y. Zhang. Accurate spectral super-resolution from single RGB image using multi-scale CNN. In Chinese Conference on Pattern Recognition and Computer Vision, pages 206–217, 2018.

[125] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[126] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. CoRR, 2015.

[127] S. W. Zamir et al. Multi-stage progressive image restoration. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14816–14826, 2021.

[128] J. Zarka, L. Thiry, T. Angles, and S. Mallat. Deep network classification by scattering and homotopy dictionary learning. In International Conference on Learning Representations (ICLR), 2020.

[129] J. Zhang et al. A survey on computational spectral reconstruction methods from RGB to hyperspectral imaging. Scientific Reports, 12(11905), 2022.

[130] L. Zhang et al. Pixel-aware deep function-mixture network for spectral super-resolution. In Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, pages 12821–12828, 2020.

[131] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar. A review of convolutional neural networks in computer vision. Artificial Intelligence Review, 57(99), 2024.

[132] Y. Zhao et al. HSGAN: Hyperspectral reconstruction from RGB images with generative adversarial network. IEEE Transactions on Neural Networks and Learning Systems, pages 1–14, 2023.

[133] Y. Zhao, L.-M. Po, Q. Yan, W. Liu, and T. Lin. Hierarchical regression network for spectral reconstruction from RGB images. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1695–1704, 2020.

[134] S. W. Zimir et al. Restormer: Efficient transformer for high-resolution image restoration. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5718–5729, 2022.

[135] C. Zou, C. Zhang, and C. Zou. Attention and transformer complementary fusion network for hyperspectral image spectral reconstruction. International Journal of Remote Sensing, 45(15):5095 – 5112, 2024.

[136] D. Zou. Nonlinear Analysis of Phase Retrieval and Deep Learning. PhD thesis, University of Maryland - College Park, 2017.

[137] D. Zou and G. Lerman. Graph convolutional neural networks via scattering. Applied and Computational Harmonic Analysis, 49(3):1046–1074, 2020.