Analysis and Reconstruction of Coherent Diffractive Imaging Using Physics Aware Deep Learning

A Thesis Proposal

Presented by

Ziqiao Guan

Department of Computer Science

Stony Brook University

September 2019
Abstract

Analysis and Reconstruction of Coherent Diffractive Imaging Using Physics Aware Deep Learning

Coherent Diffractive Imaging (CDI) is an imaging technique for probing physical structure of materials at molecular and nanoscale. In CDI, a highly coherent beam of x-rays, electrons or other forms of light source is shone through a material sample, which is scattered by the sample’s internal structure and forms a diffractive far-field pattern on an area detector. Due to its nanoscale resolution, CDI techniques can apply to studying crystal structures, nanomaterials and potentially proteins, and are of great significance in material science. Modern beamline facilities around the world are capable of generating terabytes of images daily. Due to this high volume of data, intelligent machine analysis techniques would be extremely beneficial for material science discoveries.

Today’s deep learning and convolutional neural network (CNN) methods have reached out to specialized applications that otherwise would not be adequately tackled, and have meanwhile presented new challenges. For CDI, the first question is whether we choose to investigate in real space or image space (reciprocal space) — CDI is essentially a phase-less Fourier transform of the structure of material samples (in real space), and thus reconstructing material structure means recovering the lost phase, which could be expensive. Otherwise, should we circumvent the reconstruction problem by directly analyzing images in reciprocal space, there are two main challenges for CNN learning: (1) not enough labeled data, and (2) weak, scattered and noisy features with background interference. Whether it is a discriminative model or generative model, in real space or reciprocal space, we need to incorporate domain knowledge in the learning process, so that the captured features are aware of our physics understanding and consistent with scientists’ interest, which leads to better prediction accuracy, output quality and decision making in experiments.

In this proposal, we first present a novel joint model in image space — Double-View Fourier-Bessel Convolutional Neural Network (DVFB-CNN) to perform multi-label image annotation. For encoding the common structural symmetry in x-ray scattering images, We articulate a physics-aware image feature transform, Fourier-Bessel transform (FBT), in conjunction with deep representation learning, to perform learning in both image domain and polar frequency domain. For polar frequency analysis, we develop an FBT estimation algorithm for partially observed x-ray images, and train a dedicated CNN to extract structural information from FBT. We demonstrate that our deep Fourier-Bessel features well complement standard convolutional features, and the joint network (i.e., DVFB-CNN) improves mean average precision by 13% in multi-label annotation. We also conduct transfer learning on real experimental
Identifying weak scattered patterns with diffuse background interference is an open challenge in training CNNs that arises in scientific imaging. In order to enhance feature representations for those tricky patterns in image space, we articulate an Attentional Aggregation Module (AAM). First, we reweight and highlight important features in the images using data-driven attention maps. We decompose the attention maps into channel and spatial attention components. In the spatial attention component, we design a mechanism to generate multiple spatial attention maps tailored for diversified multi-label learning. Then, we condense the enhanced local features into non-local representations by performing feature aggregation. Both attention and aggregation are designed as network layers with learnable parameters so that CNN training remains fluidly end-to-end, and we apply it in-network a few times so that the feature enhancement is multi-scale. We conduct extensive experiments on CNN training and testing, as well as transfer learning, and empirical studies confirm that our method enhances the discriminative power of visual features of scientific imaging.

For the generative approach in real space, we present PtychoNet, a deep learning based method to perform phase retrieval for ptychography. Ptychography is a CDI method that captures multiple diffraction patterns of a sample with a set of shifted localized illuminations (“probes”). The reconstruction problem, known as “phase retrieval”, is typically solved by iterative algorithms. We devise a generative network to encode a full ptychography scan, reverse the diffractions at each scanning point and compute the amplitude and phase of the object in a non-iterative manner. We demonstrate successful reconstructions using PtychoNet as well as recovering fine features in the case of extreme sparse scanning where conventional iterative methods fail to give recognizable features.

We also address the issue of learning from small dataset using partial differential equations (PDEs) guided diffusion. We present a human-oriented feature detection framework, Learning Diffusion on Global Graph (LDGG), to understand personalized interests in 3D models, given a small set of training and testing models. We outline three ongoing work on CDI and physics-aware learning, including PtychoNet interpretations, flexible ptychography reconstruction and fluorescent image super-resolution.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Statement</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Challenges</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Organization of Thesis Proposal</td>
<td>3</td>
</tr>
<tr>
<td><strong>2 Background</strong></td>
<td>5</td>
</tr>
<tr>
<td>2.1 Convolutional Neural Network (CNN)</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Batch Normalization</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2 Residual Learning</td>
<td>7</td>
</tr>
<tr>
<td>2.1.3 Adversarial Learning</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Diffraction Image Analysis</td>
<td>9</td>
</tr>
<tr>
<td>2.2.1 Clustering</td>
<td>10</td>
</tr>
<tr>
<td>2.2.2 Hand-crafted Descriptors</td>
<td>12</td>
</tr>
<tr>
<td>2.2.3 Deep Learning Classification</td>
<td>13</td>
</tr>
<tr>
<td>2.3 The Phase Retrieval Problem</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Error Reduction and Alternating Algorithms</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Descent Algorithms</td>
<td>17</td>
</tr>
<tr>
<td>2.3.3 Ptychography and Phase Retrieval</td>
<td>17</td>
</tr>
<tr>
<td><strong>3 Double-View Image Feature with Fourier-Bessel Transform</strong></td>
<td>21</td>
</tr>
<tr>
<td>3.1 Introduction and Motivation</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Related Work</td>
<td>22</td>
</tr>
<tr>
<td>3.3 Background, Challenges, and Strategies</td>
<td>23</td>
</tr>
<tr>
<td>3.3.1 Transfer Learning with Synthetic Dataset</td>
<td>24</td>
</tr>
<tr>
<td>3.3.2 Fourier-Bessel Transform</td>
<td>25</td>
</tr>
</tbody>
</table>
3.4 Double-View Fourier-Bessel Convolutional Neural Network . . . . . . 26
  3.4.1 Fourier-Bessel Estimate of Partial Images . . . . . . . . . . . . 27
  3.4.2 Double-View Convolution . . . . . . . . . . . . . . . . . . . 27
  3.4.3 Training . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.5 Experiments and Evaluations . . . . . . . . . . . . . . . . . . . . . 28
  3.5.1 End-to-end Training and Testing: Synthetic Dataset . . . . . . 29
  3.5.2 Transfer Learning: Experimental Dataset . . . . . . . . . . . . 30
3.6 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

4 Image Feature Enhancement with Attentional Aggregation 35
  4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
  4.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
  4.3 Attentional Aggregation Module . . . . . . . . . . . . . . . . . . . . 38
    4.3.1 Attention: Local Refinement . . . . . . . . . . . . . . . . . . . 40
    4.3.2 Aggregation: Non-local Representation . . . . . . . . . . . . . 41
    4.3.3 Network Architecture . . . . . . . . . . . . . . . . . . . . . . . 42
  4.4 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
    4.4.1 Datasets and Metrics . . . . . . . . . . . . . . . . . . . . . . . . 43
    4.4.2 Ablation Studies . . . . . . . . . . . . . . . . . . . . . . . . . . 43
    4.4.3 Comparison with Other Attention Methods . . . . . . . . . . . 45
    4.4.4 Transfer Learning with Experimental Dataset . . . . . . . . . 46
    4.4.5 Classification with Fashion-MNIST . . . . . . . . . . . . . . . 46
    4.4.6 Attention Visualization . . . . . . . . . . . . . . . . . . . . . . 46
  4.5 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

5 Ptychography Phase Retrieval 49
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
    5.1.1 Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
    5.1.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
  5.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
  5.3 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
  5.4 PtychoNet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
  5.5 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
    5.5.1 Implementation Details . . . . . . . . . . . . . . . . . . . . . . . 54
    5.5.2 Comparison with Iterative Algorithms . . . . . . . . . . . . . 55
    5.5.3 Tests on BrainMaps Image Dataset . . . . . . . . . . . . . . . 55
    5.5.4 Performance with Low Overlap . . . . . . . . . . . . . . . . . . 57
  5.6 Extreme Low Overlap Reconstruction . . . . . . . . . . . . . . . . . 59
  5.7 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
6 Other Work and Conclusions

6.1 Ongoing Work
6.1.1 Interpretation of PtychoNet
6.1.2 Flexible Ptychography Reconstruction
6.1.3 Fluorescent Image Super-resolution

6.2 Small Dataset Feature Learning Using PDE-guided Diffusion

6.3 Thesis Proposal Summary

Publications

Bibliography
Chapter 1

Introduction

1.1 Problem Statement

Coherent Diffractive Imaging (CDI) is an imaging technique for probing physical structure of materials at molecular and nanoscale. In CDI, a highly coherent beam of x-rays, electrons or other forms of light source is shone through a material sample, which is scattered by the sample’s internal structure and forms a diffractive far-field pattern on an area detector. CDI is widely used to study crystal structures, nanomaterials and potentially proteins, and are of great significance in material science.

With its important role in science, CDI has evolved rapidly in recent years, with many new setups developed for various application scenarios [98]. Modern x-ray detectors are capable of generating 50,000 to 1,000,000 images/day (i.e., 1-4TB/day). With all those high-throughput x-ray scattering facilities worldwide, this amount of data is beyond what human analysis can cover, and intelligent machine processing methods are desperately needed.

In the past few years, deep learning has made groundbreaking improvements in every aspect of artificial intelligence. In computer vision, convolutional neural networks (CNNs) are powering applications in object detection [37], face recognition [99], image captioning [145], autonomous driving [16] and many more. The success of CNNs comes from their extremely high complexity and capacity, and intensive training from massive amount of labeled data. CNNs have also proven to be widely applicable to a diverse variation of data, image types and tasks that give rise to endless specialized applications. Nevertheless, these applications pose new challenges with unique data distributions, and more often than not, a relatively smaller labeled dataset.

This proposal is going to cover our work on applying powerful deep learning techniques to understanding CDI, with both discriminative and generative tasks. Since the imaging is determined by the underlying physics, our main strategy is to incor-
porate physics understanding into designing and training machine learning models, so that they are better geared towards experimental physics and able to learn feature representations from smaller, more challenging datasets.

1.2 Challenges

The diffractive imaging process is essentially Fourier transform. Denote the sample’s spatial distribution as $\rho(r)$, and the diffraction image can be written as

$$I(q) = \left| \int_V \rho(r) \exp(iq \cdot r) \, dV \right|^2,$$

where only the intensity is captured. Since the phase information is lost in the process, to recover the real space $\rho(r)$ from the image space (reciprocal space) is an ill-posed problem. Conventional phase retrieval methods are based on iterative methods which perform transform between real space and reciprocal space back and forth, and correct the intensity to the known values at each step [25]. This type of methods are computationally intensive. In order to speed up this process, we need to utilize machine learning to develop smarter iterative or non-iterative reconstruction methods.

On the other hand, the reconstruction problem can be circumvented if we perform data analysis directly in the image space. This is a multi-label image annotation problem, which means the attributes are not mutually exclusive, and an image can bear multiple or none of the attributes. For CDI specifically, there are two main challenges for training CNNs. The first challenge is lack of labeled data. As much as the amount of generated data is abundant in the CDI community, the amount labeled is relatively small. This is because CDI image data would require actual domain experts to label, as opposed to crowdsourcing in natural image datasets. For example, one of our experimental dataset contains less than 3,000 images, which is far from enough to train a CNN properly. The second challenge is meaningful patterns in CDI are often weak, scattered and noisy with background interference. This does not work well with convolutions, since they are inherently local. Because of these technical challenges, we need effective methods to represent and transform the features, so that the annotations can make more reliable inferences based on the correct visual clues.

1.3 Contributions

The contributions in this proposal are mainly as follows:
• We presented a double-view Fourier-Bessel image feature that dramatically improved the ability to encode image structural information. This feature exploits the image symmetry caused by the scattering process and effectively encodes structural information. We introduce a joint feature from dual CNNs to fuse multiple views of features and an extensible framework to utilize even more, which opens up opportunities to involve more aspects of domain knowledge (physics) in the inference.

• We presented the Attentional Aggregation Module (AAM) to enhance image features both locally and globally, which is a general purpose feature enhancement method without transforms or prior knowledge of the features. From a feature space’s point of view, CDI features lie in a smaller, more “restricted” subspace due to the weak, scattered and noisy patterns. Our attention and aggregation method aims to identify this subspace to better represent the features. AAM is designed to be differentiable and in-network and it can be applied multiple times for multi-scale feature enhancement.

• We presented PtychoNet, a fast non-iterative ptychography reconstruction method. We reverse the diffractive imaging using deep generative neural network instead of the conventional iterative descent, or optimization based framework. We show that learning from data is exceptionally powerful to complete the missing phase, and meanwhile, make the drastic transition from reciprocal space back to real space in a generative model, which is a large step forward compared to tasks such as image segmentation or translation, where locality is preserved. We also show that it is possible to perform ptychographic reconstruction with a much sparser scanning pattern using PtychoNet, which calls for a major rethinking for the necessary overlap conditions previously known among the CDI community.

1.4 Organization of Thesis Proposal

The rest of the thesis proposal is organized as follows. We review the background and related techniques of our problem in Chapter 2. In Chapter 3, we present our work of a double-view Fourier-Bessel image feature. In Chapter 4, we present the Attentional Aggregation Model (AAM) to enhance image features. In Chapter 5, we present PtychoNet, a deep neural network for non-iterative ptychography reconstruction. Finally, we introduce our other work as well as our ongoing work on CDI and physics-aware learning, and draw the conclusion in Chapter 6.
Chapter 2

Background

In this chapter, we briefly review some relevant theories and models concerning deep learning and analysis of coherent diffractive imaging (CDI). First, we introduce the basics of convolutional neural networks (CNNs), as well as the important modern techniques to combat training troubles and overfitting. Then we review the key problems and methods for studying CDI, covering both discriminative (classification) and generative (reconstruction) problems, and we especially point out some methods that utilize deep CNNs.

2.1 Convolutional Neural Network (CNN)

For the last decade, the most important model for deep learning in computer vision is the convolutional neural network (CNN) \([64, 68]\). CNN is a learnable structure that normally takes image as input, computes a sequence of layers of operations, including convolution, pooling, affine transform, nonlinear activation etc., to generate some prediction output. CNN can be constructed for not only classification problems, predicting 0-1 labels, but also regression problems, and in particular, generate new images with convolution transpose \([82]\). AlexNet \([64]\) is an exemplary 5-layer CNN model that rekindled the most recent surge in deep learning computer vision research, shown in Figure 2.1.

Research proves that for CNN, being deeper is the key to better performance \([118]\). However, as the depth increases to 50, 100 and over 1,000 layers, new problems arise that cause the training and testing performance to stagnate and even worsen. This shows that the issue with CNN performance often comes with the process of parameter update in optimization, \(e.g\). gradient descent and variants. Researchers have proposed new tactics to ease these numerical troubles, and we will report some of the most influential tactics in this section.
**Figure 2.1:** AlexNet is a 5-layer CNN for image classification. It takes a $227 \times 227$ image as input, and computes a 1000-D vector to predict the correct object category for the image out of 1000 categories. In the network, convolutional layers (Conv) and fully connected (FC) layers are followed by ReLU. Max pooling layers (Maxpool) downsample the features by spatially pooling the maximum in $2 \times 2$ windows. The final FC8 is followed by a softmax to output probabilities.

### 2.1.1 Batch Normalization

Like other statistical learning methods, it is important to normalize the input features when training a CNN. Normalization, also known as whitening or standardization, scales and shifts the feature to a standard distribution in order to avoid numerical overflow or underflow and improve model accuracy. In deep CNNs, however, shifts in feature distributions can also be caused by computations through the layers. This leads to unstable internal feature distributions each time the model parameters get updated, counteracting the training step, which is called internal covariate shift.

Batch Normalization [52] (BN) is a mechanism to reduce internal covariate shift in deep neural network training. It is designed to normalize the features in the internal layers and meanwhile account for the optimization step, so that the model parameters would not blow up as gradient descent counters the normalization. The $d$-dimensional feature to normalize $\mathbf{x} = (x^{(1)}, \ldots, x^{(d)})$ will be normalized at each dimension

$$
\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}},
$$

(2.1)

to have zero mean and unit variance, with mini-batch statistics, hence the name.

BN consists of a normalization and an affine transform. For a batch $B = \{x_1, \ldots, x_m\}$, the input feature is first normalized with mini-batch mean and variance:

$$
\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}},
$$

(2.2)
2.1 Convolutional Neural Network (CNN)

Figure 2.2: A basic residual block.

where $\mu_B$ is batch mean, $\sigma^2_B$ is batch variance, and $\epsilon$ is a small constant. During training, $\mu_B$ and $\sigma^2_B$ are updated with running average and variance algorithms and saved along with the trained model parameters. During evaluation, they are fixed to the saved values in order to guarantee that training and testing data have identical distribution. Then the normalized values are scaled and shifted:

$$y_i = \gamma \hat{x}_i + \beta = \text{BN}_{\gamma,\beta}(x_i),$$  \hspace{1cm} (2.3)

where $\gamma$ and $\beta$ are learned parameters. The affine transform ensures that BN can represent identity transform (i.e., BN can be reverted), and therefore transform the data to arbitrary range of the nonlinear activation for more flexible feature representation.

2.1.2 Residual Learning

Very deep CNNs are known to be afflicted by a degradation problem — with the network depth increasing, accuracy gets saturated and then degrades rapidly. However, this degradation is not caused by overfitting, since it is the training error that would start to grow, whereas overfitting typically causes the training error to decrease, but the testing performance to worsen. By construction, a deeper network with added layers shall be no worse than the original, considering the added layers can simply be identity mapping. Following this thought, deep residual network \cite{HeK2015} is proposed to explicitly fit a residual mapping instead.

Formally, denote the desired mapping as $\mathcal{H}(x)$. The residual mapping to fit can be written as

$$\mathcal{F}(x) = \mathcal{H}(x) - x,$$  \hspace{1cm} (2.4)

which can later added back to acquire $\mathcal{H}(x) = \mathcal{F}(x) + x$. Essentially, this gives a formulation with “shortcut connections”, where a shortcut of identity mapping bypasses certain layers and adds back to their output. A basic residual block is shown in Figure 2.2. Residual learning is simple to construct and empirically effective, and the
idea of residual bypass is adapted to other networks, e.g., DenseNet [46], U-Net [108], Hourglass network [90], ResNeXt [138] and residual attention network [132].

The cascading residual blocks are also inherently related to multi-scale analysis, as the identity mapping can be viewed as overall structure, while the residual is finer scale detail. In that sense, residual learning can encourage better separation of multi-scale features and form scale sensitive feature representations.

2.1.3 Adversarial Learning

Deep neural networks are extremely powerful in discriminative problems, where the task is to classify the input with some class labels. Deep generative models, on the other hand, are harder to learn, because the output is much more high-dimensional and sophisticated, and common loss functions do not represent the distance in the output space properly. For example, a network that generates new images (e.g., for segmentation) is often inadequately trained with mean square error (MSE), or L2 loss. Minimizing a sum of squared pixel errors easily pushes the output to a local minimum of some uniform average colors. As a result, the output usually ends up blurry.

Generative Adversarial Networks (GANs) [32] address this difficulty by formulating two network adversaries: a generator $G$ and a discriminator $D$. Assume that real data $x$ is drawn from the data distribution $p_{data}$. We train $G$ to generate a new sample from an input noise $z$: $G(z) \sim p_g$. Meanwhile, $D$ is trained to tell whether a sample is from real data distribution (real) or generated (fake). The two networks are trained simultaneously in the form of a two-player minimax game with value function $V(D,G)$:

$$
\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))].
$$

(2.5)

In other words, the generator is trained to fool the discriminator with fake samples, and the discriminator is trained to accurately distinguish real and fake. The generator $G$ is able to generate samples that better emulate $p_{data}$ due to adversarial training.

GANs are difficult to train and prone to “mode collapse”, i.e. having all generator outputs reduce to a single mode (cluster) in the data distribution. Various extensions and improvements are proposed on top of the original GAN:

- **Wasserstein GAN** (W-GAN) [4] utilizes Earth Mover’s Distance (EMD), a weaker distance, to form a continuous and differentiable loss for the discriminator, so that the loss can steadily decrease even when $p_g$ and $p_{data}$ are disjoint. W-GAN uses a ‘critic’ function $f$ instead of the discriminator $D$ and requires
2.2 Diffraction Image Analysis

Analyzing x-ray diffraction images is a topic with connections to experimental physics and challenges in computer vision. Researchers have proposed various methods, including unsupervised and supervised learning methods, and both methods using hand-crafted descriptors and deep CNNs.

\( f \) to be \( K\)-Lipschitz for some \( K \), which is enforced by weight clipping. The objective of W-GAN is written as

\[
\min_g \max_f \{E_{x \sim p_{\text{data}}}(x)[f(x)] - E_{z \sim p_z(z)}[f(g(z))]\}. \tag{2.6}
\]

- **Conditional GAN (cGAN)** conditions the discriminator with additional information, such as labels, text or images so that the network learns the mapping between the given information and output. It is particularly effective to solve image-to-image translation problems \[53\], where the objective of cGAN is written as

\[
\min_G \max_D \{E_{x,y}[\log D(x, y)] + E_{x,z}[\log(1 - D(x, G(x, z)))]\}, \tag{2.7}
\]

where \( x \) is the input (source) image, \( y \) is the output (target) image, and \( z \) is an input noise.

- **CycleGAN \[149\]** removes the need for paired training data in image-to-image translation problems. For the two domains of source and target images, \( X \) and \( Y \), CycleGAN trains a pair of mappings \( F : X \rightarrow Y \) and \( G : Y \rightarrow X \) to perform translations in both directions. They are trained with discriminators \( D_X \) and \( D_Y \), and the adversarial loss is expressed as

\[
\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = E_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + E_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))], \tag{2.8}
\]

\[
\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = E_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + E_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y)))]. \tag{2.9}
\]

Additionally, cycle consistency loss is enforced:

\[
\mathcal{L}_{\text{cyc}}(G, F) = E_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + E_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]. \tag{2.10}
\]

The full objective is

\[
\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F), \tag{2.11}
\]

\[
G^*, F^* = \arg \min_{G,F} \max_{D_X,D_Y} \mathcal{L}(G, F, D_X, D_Y). \tag{2.12}
\]
2.2.1 Clustering

**Spectral Clustering.** Yoon et al. [144] used spectral clustering to classify x-ray diffraction images. In contrast to standard $K$-means clustering, which perform the clustering in the Euclidean space, spectral clustering acts on the graph Laplacian basis [88] and captures the nonlinear correlations of the images.

First, a graph of the image data $G = (V, E)$ is constructed to represent the dataset. Each vertex $v_i \in V$ represents a $n$-D data point, here one single image, and edge $e_{ij} \in E$ connects two vertices $v_i$ and $v_j$, indicating they are similar. The edges here are binary, or unweighted, meaning $e_{ij}$ is either 1 if $v_i$ and $v_j$ are connected, or 0 otherwise. Thus, $G$ forms a similarity graph of the images. The unweighted affinity matrix is defined as $W = (e_{ij})_{i,j}$ and the degree matrix is $D = \text{diag}(\sum_j e_{ij})_{i}$, where each diagonal entry represents the number of edges connecting a vertex.

$W$ can be constructed in different ways to reflect pairwise image similarity [130], e.g., mutual $k$-nearest neighbor graph (MKNN). In MKNN, vertices $v_i$ and $v_j$ are connected if and only if $v_i$ is among the $k$ nearest neighbors of $v_j$ in Euclidean distance, and vice versa. This construction is able to adapt to varying density of data point distributions.

The graph Laplacian is constructed from $W$. It can be either unnormalized or normalized as follows

\[
(\text{Unnormalized}) \quad L = D - W, \quad \text{(2.13)}
\]

\[
(\text{Normalized}) \quad L_{\text{norm}} = I - D^{-1} W. \quad \text{(2.14)}
\]

Here normalized graph Laplacian is used. We then solve the eigenvalue problem:

\[
L_{\text{norm}} \psi = \lambda \psi, \quad \text{(2.15)}
\]

to find the eigenvalues $\{\lambda_i\}$ and eigenvectors $\{\psi_i\}$. We sort $\{\lambda_i\}$ in ascending order

\[0 \leq \lambda_0 \leq \lambda_1 \leq \ldots,\]

and let

\[
\Psi = (\psi_0, \ldots, \psi_{m-1}), \quad \text{(2.16)}
\]

where the $i$-th column is the $i$-th largest eigenvector of $L_{\text{norm}}$. $\Psi$ forms the truncated Manifold Harmonic Basis (MHB) [127] of $G$, and the $j$-th row of $\Psi$, denoted as $\Psi'_j$, represents the truncated coordinate of data point $v_j$ in the MHB. Finally, we perform $K$-means clustering on $\{\Psi'_j\}$ and the result is the spectral clustering of images $\{v_j\}$. The overall algorithm to perform spectral clustering is shown in Algorithm 1.

**Diffusion-based Clustering.** Huang et al. [49] incorporated two new techniques into the spectral clustering framework — Earth Mover’s Distance (EMD) and Aggregated Heat Kernel (AHK), in order to further improve the identification of informative features under noisy and heterogeneous density distributions.
ALGORITHM 1: Spectral Clustering.

**Input:** Set of data points \( V \), number of clusters \( m \).

**Output:** Cluster assignments of all \( v_i \).

1. Construct the affinity matrix \( W \) from the vertices \( V \);
2. Compute the diagonal matrix \( D \) where \( D_{ii} = \sum_j w_{ij} \) and \( D_{ij} = 0 \) for \( i \neq j \);
3. Compute the graph Laplacian \( L \) using (2.13) or (2.14);
4. Compute the first \( m \) eigenvalues of \( L \) and put them in columns of \( \Psi \) as in (2.16);
5. Run \( K \)-means with the rows of \( \Psi \) to compute \( m \) clusters;

Earth Mover’s Distance (EMD) is derived in the context of optimal mass transport (OMT) problems \[109\]. It is defined as the minimum amount of work to transform one distribution (i.e., mass) to another, computed by the amount of transported earth multiplied by the unit cost from source to destination. Formally, for two distributions \( P_r \) and \( P_g \), EMD, or Wasserstein-1 distance, is defined as

\[
W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbf{E}_{(x, y) \sim \gamma}[\|x - y\|],
\]

where \( \Pi(P_r, P_g) \) represents the set of all joint distributions \( \gamma(x, y) \) that have marginal distributions of exactly \( P_r \) and \( P_g \). EMD is a natural distance for matching, since it explicitly represents the transformation effort from one distribution to another, whether they are similar or dissimilar.

Computing EMD can be formulated as a linear programming problem. Here, we denote the first image as \( P = [(p_1, w_{p_1}), \ldots, (p_m, w_{p_m})] \), where \( p_i \) is the \( i \)-th row of the image, and \( w_{p_i} \) is the weight of the row. Similarly, the second image is represented as \( Q = [(q_1, w_{q_1}), \ldots, (q_n, w_{q_n})] \). Let \( D = [d(i, j)]_{i,j} \) be the distance matrix where \( d(i, j) \) is the distance between \( p_i \) and \( q_j \), e.g. 2-norm distance. EMD is equivalent to finding a flow \( F = [f(i, j)]_{i,j} \), where \( f(i, j) \) is the flow from \( p_i \) to \( q_j \) that minimizes the total cost:

\[
\text{WORK}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(i, j)d(i, j),
\]

s.t. \( f(i, j) \geq 0 \), \( 1 \leq i \leq m, 1 \leq j \leq n \),

\[
\sum_{j=1}^{n} f(i, j) \leq w_{p_i}, \quad 1 \leq i \leq m,
\]

\[
\sum_{i=1}^{m} f(i, j) \leq w_{q_j}, \quad 1 \leq j \leq n,
\]

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} f(i, j) = \min \left( \sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j} \right).
\]
Once the optimal flow $\mathbf{F}$ is found, EMD is subsequently computed as

$$E(\mathbf{P}, \mathbf{Q}) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(i,j)d(i,j) \over \sum_{i=1}^{m} \sum_{j=1}^{n} f(i,j).$$ (2.19)

Based on EMD, we can define the Earth Mover’s Similarity (EMS) as

$$W_{EMS}(\mathbf{P}, \mathbf{Q}) = \exp \left( - \frac{E(\mathbf{P}, \mathbf{Q})}{2\sigma^2} \right).$$ (2.20)

Aggregated Heat Kernel (AHK) [50] is defined as the integration of the heat kernel over time, and it is inherently related to Green’s function. Intuitively, AHK indicates the amount of heat diffusion during a period of time, from immediate neighborhood to an extended region. Due to its multi-scale nature, AHK can be applied to robust spectral clustering. Given an affinity matrix $\mathbf{W}$ and its MHB $\{(\lambda_i, \psi_i)\}_i$ from (2.16), AHK can be written as

$$W_{AHK}(i,j) = \sum_k \left[ \frac{1}{\lambda_k + \gamma} \psi_k(i)\psi_k(j) \right],$$ (2.21)

where $\gamma$ is a smoothing factor.

To perform EMS-AHK clustering, we first compute the AHK matrix $\mathbf{W}_{AHK}$ using (2.21) from the affinity matrix $\mathbf{W}_{EMS}$ with pairwise EMS. Finally, we compute the eigen decomposition $\{(\mu_i, \phi_i)\}_i$ of $\mathbf{W}_{AHK}$ and perform spectral clustering using $\mathbf{W}_{AHK}$.

### 2.2.2 Hand-crafted Descriptors

For supervised learning, the traditional method is to first construct some form of feature representations of the data, and then train a classifier to classify the features with the labels. Kiapour et al. [61] reported a selection of features that are widely used in computer vision:

- **Downsampled (tiny images)** [126], downsizing the images to $32 \times 32$.
- **GIST** [95], a low dimensional abstract representation of the image. GIST first computes convolutions of the image with multi-scale, multi-orientation Gabor filters and spatially averages the result in a $4 \times 4$ grid.
- **Histogram of Oriented Gradients (HOG)** [17] encodes local shape information by computing the distributions of gradient orientations in local regions.
• Pyramid of HOG (PHOG) \cite{10} captures multiple scales of HOG.

• Local Binary Patterns (LBP) \cite{94} encodes the local structure with an 8-digit binary code, indicating the relations between a central pixel and its 8 neighbors, and constructs a feature vector from the binary codes using histogram binning and concatenation.

• SIFT \cite{83} is computed and then aggregated using bag-of-words.

Linear SVM is used to classify the image attributes. Other statistics learning model can also be learned, \textit{e.g.} kernel SVM, gradient boosting or random forest.

Some attributes are closely related to a certain group of samples, \textit{e.g.} small-angle x-ray scattering (SAXS) or wide-angle x-ray scattering (WAXS). In this case, we can build a two-level classification model. At the top level, the classifier predicts the group to which the image sample belongs (SAXS \textit{vs.} WAXS), and then two different sets of SVM classifiers are trained to specialize in the two groups. This can also extend to a hierarchical classification model, where the attributes from general to fine-grained (\textit{e.g.}, Ring \textit{vs.} Ring: Isotropic) are predicted in a set of classifiers organized as a decision tree.

Another important technique of feature construction is feature ensemble. In experiments, LBP and PHOG have the best performance when predicting the X-ray Materials Discovery dataset (XMD). LBPHOG descriptor concatenates the feature vectors of LBP and PHOG and achieves better average precisions (APs) than the two individual descriptors.

### 2.2.3 Deep Learning Classification

Researchers have been attempting to bring the success of CNN in natural image and video data to more specific, dedicated datasets and applications. One application that is very close to our problem is Yann and Tang’s CrystalNet \cite{143}, using CNN to classify x-ray protein crystallization images. They proposed a 4-layer convolutional network for classification. Their results show that CNNs can adapt to the x-ray image dataset and outperform other statistical learning methods. However, there are 2 key distinctions between CrystalNet and classification in CDI — first, their dataset is x-ray microscopy images with no diffraction, so the images contain real space objects, which are easier to encode compared to diffractive images that are more diffused and scattered due to Fourier transform; second, CrystalNet performs multi-class classification, \textit{i.e.} each sample belongs to one out of \(N\) classes, while in our applications the classification problem is multi-label, meaning the attributes are not mutually exclusive, and a sample can bear multiple or none of the attributes. As
a result, CDI analysis calls for more in-depth study of the intricate correlations of both the image space and the label space.

**1D Convolutions.** Park et al. proposed a 1D CNN as an attempt to manipulate the image space in CDI [97]. Their method utilizes a 1D input data — the circular average curve computed from the x-ray diffraction images, because intensity in the radial direction is informative to deduce image properties. The CNN contains 3 convolutional layers. After pooling the last convolutional layer, the network splits into 3 classifiers, each of which is made of fully connected (FC) layers, to predict space groups, extinction groups and crystal systems respectively. The network architecture is shown in Figure 2.3

**Convolutional Autoencoder.** Autoencoders are capable of extracting features in an unsupervised manner by enforcing a reconstruction of the input through multiple
2.3 The Phase Retrieval Problem

Reconstruction of CDI is essentially recovering the lost phase from the imaging (1.1), and thus it is called phase retrieval. It can be formulated as an optimization problem:

$$\min_A \| \mathcal{F}A \|^2 - I \|^2, \quad (2.22)$$

where $I$ is the captured image, and $A$ is the computed sample in real space. The objective of the optimization is to find the real space distribution to match the measured Fourier intensity.

Generally, phase retrieval is an ill-posed problem because the missing phase has infinite possibilities. A successful reconstruction requires proper prior and regularization (e.g., support in real space, non-negative constraint). Conventional phase retrieval algorithms mainly fall into 2 categories: one is alternating style, usually

---

**Figure 2.4:** Architecture of convolutional autoencoder proposed by Wang et al.. Image courtesy of [131].

layers. Normally, a middle layer is set to have fewer numbers of parameters than the input dimension, which is called bottleneck layer, and thereby force a compact encoding of the input. Convolutional autoencoder [86] has been used in medical imaging [43] to pretrain a CNN in order to improve the feature extraction, given the dataset is limited and a full supervised training is not good enough.

Wang et al. proposed to include a convolutional patch autoencoder [131] as part of the feature ensemble for x-ray scattering image classification. They choose to train the autoencoder with randomly sampled image patches instead of full images in order to preserve more details in the large images, as well as create more available patches for training. In the patch autoencoder, a softmax is put in the middle layer to cluster the image patches. The network architecture is shown in Figure 2.4. For feature extraction, the softmax layer is taken as feature vector and they use spatial pyramid matching (SPM) [66] to generate 3 scales of autoencoder features from coarse to fine and perform sum pooling on those features.
Background

involving enforcing real space and image space constraints alternately; another is
descent style, using gradient descent and variant methods to attack the optimiza-
tion objective. These two types of methods are both iteration based and inherently
connected. In this section, we will explain some typical methods in more details.

2.3.1 Error Reduction and Alternating Algorithms

The Gerchberg-Saxton algorithm was invented by Gerchberg and Saxton [27] to solve
the phase reconstruction problem of two intensity measurements, i.e. only the two
intensities in real space and image space are known. Formally, denote the object (real
space) as

$$ f(x) = |f(x)| \exp[i\eta(x)], $$

where $\eta(x)$ is the phase of the object; its Fourier transform

$$ F(u) = |F(u)| \exp[i\psi(u)] = \mathcal{F}[f(x)]. $$

For applications in image processing, we assume $f(x)$ and $F(u)$ are both $n \times n$ arrays.
The algorithm consists of iterating over the following 4 steps: (1) Fourier transform
the object estimate; (2) replace the amplitude of the computed transform with that of
measured $F(x)$; (3) inverse transform the replaced estimate; (4) replace the amplitude
of the computed inverse transform with that of measured $f(x)$ as the estimated object
in a new round of iteration. It can be written in expressions as follows:

$$ G_k(u) = |G_k(u)| \exp[i\phi_k(u)] = \mathcal{F}[g_k(u)], $$

$$ G'_k(u) = |F(u)| \exp[i\phi_k(u)], $$

$$ g'_k(x) = |g'_k(x)| \exp[i\theta'_k(x)] = \mathcal{F}^{-1}[G'_k(u)], $$

$$ g_{k+1}(x) = |f(x)| \exp[i\theta_{k+1}(x)] = |f(x)| \exp[i\theta'_k(x)], $$

where $g_k, \theta_k, G'_k, \phi_k$ are estimates of $f, \eta, F,$ and $\psi$ respectively. The G-S algorithm
can be generalized to a class of problems where partial constraints are known in the
two domains. Alternating algorithms can be applied to transform back and forth,
satisfying the constraints in between. This is often referred to as error reduction
algorithm, since the error can be rigidly proven to decrease at each iteration [25].

In CDI, the phase retrieval problem has a single intensity measurement, i.e. only $|F(u)|$ is measured in experiment. In this case, the first 3 steps are identical to that of G-S algorithm, and the fourth step is given by

$$ g_{k+1}(x) = \begin{cases} 
    g'_k(x), & x \not\in \gamma, \\
    0, & x \in \gamma,
\end{cases} $$

where $\gamma$ is the set of points where object-domain constraints are violated, i.e. where $g'_k(x)$ is negative, or nonzero outside of the known support.
2.3 The Phase Retrieval Problem

2.3.2 Descent Algorithms

The phase retrieval problem can also be solved by directly solving the optimization (2.22) using gradient descent style algorithms. Considering $G'_k(u)$ is computed from $G_k(u)$ using (2.26) to enforce the image space (Fourier domain) constraints, we can define the squared error in image space as

$$B_k = E^2_{F_k} = \frac{1}{n^2} \sum_u |G_k(u) - G'_k(u)|^2 = \frac{1}{n^2} \sum_u [\|G_k(u) - |F(u)|\|^2].$$  \hfill (2.30)

[25] shows that for the problem of a single intensity measurement, the partial derivative of $B$ with respect to a value at a given point $g(x)$ can be reduced to

$$\partial_g B = 2[g(x) - g'(x)].$$  \hfill (2.31)

To minimize $B_k$, at the $k$-th step $g(x)$ is set to $g'_k(x)$. Therefore, the error reduction algorithm is identical to a double-length step gradient descent. Gradient descent can also be replaced by other descent algorithms, for example, conjugate gradient.

The advantage of descent style algorithms is that they directly address the optimization problem, so that the imaging formulations and constraints can be directly modified. This will be further discussed next in ptychographic phase retrieval regarding a class of maximum likelihood based methods.

2.3.3 Ptychography and Phase Retrieval

Ptychography is a CDI method that captures multiple diffraction patterns of an object with a set of shifted localized illuminations. These overlapping shifted captures provide redundancy of object measurement, so that the imaging can achieve higher resolution and cover larger areas of the objects of interest. Formally, it can be expressed as

$$I_j(q) = |\mathcal{F}[P(r - r_j)O(r)]|^2,$$  \hfill (2.32)

where $P$ is an illumination pattern, also called “probe”, and $O$ is the real space object. If we consider the Fourier transform only before the squared norm takes place, the diffraction alone is generating an exit-wave

$$\chi_j(r) = \mathcal{F}[P(r - r_j)O(r)],$$  \hfill (2.33)

and the reconstruction problem can be formulated as finding the best $P$ and $O$ that give the exit-wave closest to $\chi_j$.

Ptychographic phase retrieval problems are intuitively better constrained due to redundancy and the common factor $P$ in all the images. There are early methods like
Wigner-distribution deconvolution (WDD) \cite{6} that are non-iterative, but their memory consumption is extremely high and they are susceptible to noise. The mainstream methods are iterative as in general CDI. Here we will elaborate a few algorithms that are unique to ptychographic reconstruction, both alternating style and descent style.

**Extended ptychographic engine (ePIE)** \cite{84} is an extension of ptychographic iterative engine (PIE) \cite{24, 107}, which is an alternating algorithm for ptychography. The original PIE assumes the probe $P$ is known, while ePIE reconstructs the probe $P$ and the object $O$ simultaneously. The ePIE starts with initial guesses $O_0(r)$ and $P_0(r)$. In one iteration, the diffraction patterns are processed in a random sequence $s(j)$. Beginning with pattern $s(0)$, a guess at the exit-wave $\chi_j$ is first computed

$$
\psi_j(r) = O_j(r)P_j(r - r_{s(j)}).
$$

(2.34)

Similar to the error reduction algorithm, we then compute the Fourier transform, replace the intensity with measured value, and compute the inverse transform

$$
\Psi_j(u) = \sqrt{I_{s(j)}(u)} \frac{\mathcal{F}[\psi_j(r)]}{\mathcal{F}[\psi_j(r)]},
$$

(2.35)

$$
\psi'_j(r) = \mathcal{F}^{-1}[\Psi_j(u)].
$$

(2.36)

Then the new estimates of $O$ and $P$ are computed with a pair of update functions given by

$$
O_{j+1}(r) = O_j(r) + \alpha \frac{P^*_j(r - r_{s(j)})}{|P_j(r - r_{s(j)})|^2_{\text{max}}} (\psi'_j(r) - \psi_j(r)),
$$

(2.37)

$$
P_{j+1}(r) = P_j(r) + \beta \frac{O^*_j(r + r_{s(j)})}{|O_j(r + r_{s(j)})|^2_{\text{max}}} (\psi'_j(r) - \psi_j(r)),
$$

(2.38)

where $\alpha, \beta$ are update step sizes. This process is repeated with the rest of the patterns in the sequence $s(1), s(2), \ldots$, which forms a single ePIE iteration.

The ePIE algorithm is essentially an alternating algorithm which has its replacement step in real space substituted by two updates, for $O$ and $P$ respectively, and the two update functions are symmetrical as they update the two factors of (2.34). The factor $P^*/|P|^2_{\text{max}}$ is meant to apply larger update sizes to $O$ where $P$ is large, and therefore there is higher confidence for the update, and vice versa.

**Difference map algorithm** \cite{123, 124} formulates the phase retrieval problem as the search for the intersection point between two constraint sets \cite{22}. Here we define the state vector as

$$
\psi = (\psi_1(r), \psi_2(r), \ldots, \psi_N(r)),
$$

(2.39)
2.3 The Phase Retrieval Problem

where $\psi_j$ is defined in (2.34). From the perspective of the solution space, this state lies in a high dimensional space — the direct product of the spaces of each individual diffraction image, or “view”, and the typical operations in phase retrieval, e.g. replacing Fourier intensity and updating $O$ and $P$, constitutes the distance-minimizing projections onto the constraint sets.

The first projection $\Pi_F$, associated to the Fourier constraint, replaces the Fourier intensity of the guesses $\psi_j$ with measured values, as done in (2.37) (2.38), is denoted by

$$\Pi_F(\psi) : \psi_j \rightarrow \psi_j^F = p_F(\psi_j).$$

(2.40)

The second projection $\Pi_O$ concerns finding $\hat{O}$ and $\hat{P}$ that minimize

$$\|\psi - \psi^O\|^2 = \sum_j \sum_r |\psi_j(r) - \hat{P}(r - r_j)\hat{O}(r)|^2,$$

(2.41)

and the projection is

$$\Pi_O(\psi) : \psi_j \rightarrow \psi_j^O(r) = \hat{P}(r - r_j)\hat{O}(r).$$

(2.42)

Setting the derivatives of (2.41) to zero gives

$$\hat{O}(r) = \frac{\sum_j \hat{P}^*(r - r_j)\psi_j(r)}{\sum_j |\hat{P}(r - r_j)|^2},$$

(2.43)

$$\hat{P}(r) = \frac{\sum_j \hat{O}^*(r + r_j)\psi_j(r)}{\sum_j |\hat{O}(r + r_j)|^2}.$$

(2.44)

When the probe $P$ is known, (2.43) gives the optimal $\hat{O}$ for $\Pi_O$. However, if both $O$ and $P$ needs to be reconstructed, (2.43) and (2.44) cannot be decoupled analytically. Given the iterative algorithm, they can use $O$ and $P$ from the previous iteration as reasonable guesses, except for the first iteration where an initial probe is required. With the two projections computed, the iterative update of $\psi$ is given by

$$\psi_{n+1} = \psi_n + \Pi_F[2\Pi_O(\psi_n) - \psi_n] - \Pi_O(\psi_n).$$

(2.45)

Maximum likelihood algorithms have the flexibility to handle uncertainties in CDI, e.g. noise or perturbation of the probe. In principle, this type of algorithms models the uncertainties with statistical models, and represents the optimization objective with a joint probability of all the images $\{I_j\}$. 
Thibault and Guizar-Sicairos [125] proposed a maximum likelihood method to account for the counting statistics, a type of noise which is typical in x-ray diffractive imaging. Given a pair of guesses $O$ and $P$, the number of photons measured $n(q)$ is subject to a Poisson distribution

$$p(n_j(q)|O, P) = \frac{\tilde{I}_j(q|O, P)^{n_j(q)}}{n_j(q)!} \exp(-\tilde{I}_j(q|O, P)), \quad (2.46)$$

where $\tilde{I}_j(q|O, P) = \|F[P(r - r_j)O(r)]\|^2$. Given the observed full images $\{I_j(q) = n_j(q)\}$, the negative log-likelihood function is expressed as

$$\mathcal{L} = -\log \prod_j \prod_q p(n_j(q)|O, P))$$

$$= -\sum_j \sum_q w_j(q)[n_j(q) \log I_j(q) - I_j(q) - \log(n_j(q)!)], \quad (2.47)$$

where $w_j(q)$ is a mask that is equal to 1 for valid pixels only and to 0 for unmeasured regions and bad pixels. Minimization of (2.47) can be found by gradient descent of $O$ and $P$. This method can also incorporate Gaussian noise with (2.46) replaced accordingly.

Another optimization algorithm proposed by Guizar-Sicairos and Fienup [36] aims to address the issue of transverse translation diversity, where the lateral displacement of the object (in relation to the probe) can be inexact. Instead of a fixed set of $\{r_j\}$, they introduce a set of random variables $\{\hat{r}_j\}$. The squared error metric can be defined as

$$\mathcal{E} = \sum_j \sum_q w_j(q) \left\{ \left[\|\hat{F}_j(q)\|^2 + \delta \right]^{\gamma} - [I_j(q) + \delta]^{\gamma} \right\}^2, \quad (2.48)$$

where

$$\hat{F}_j(q) = F[O(r - \hat{r}_j)P(r)], \quad (2.49)$$

$\delta$ is a small constant for numerical precautions, $w_j$ is a mask. This is an optimization problem that involves updates of the object $O$, the probe $P$ and refinement of probe positions $\{\hat{r}_j\}$. In practice, only $O$ is updated for the first few iterations (assuming reasonable initial guesses of $P$ and $\{\hat{r}_j\}$), and the full optimization follows in subsequent iterations.
Chapter 3

Double-View Image Feature with Fourier-Bessel Transform

In this chapter, we present a double-view image feature for annotating x-ray scattering images using Fourier-Bessel Transform (FBT). The purpose of our new feature is to exploit centrosymmetry, which is caused by the x-ray diffraction imaging process. We showed that this feature improved the multi-label annotation performance compared to plain convolutional features.

3.1 Introduction and Motivation

X-ray scattering is an imaging technique for probing physical structure of materials at molecular and nano-scale. It is used for various applications such as determining protein conformations. The technique consists of shining a bright, collimated x-ray beam through a material of interest; coherent interference between x-rays scattered from internal structures casts a complex far-field pattern on an area detector. Material scientists established physical models for scattering patterns [142] so that we can deduce information about the size, orientation, and packing of atoms, molecules, and nano-domains [21, 134] from image analysis.

Modern x-ray detectors are capable of generating 50,000 to 1,000,000 images/day (i.e., 1-4 TB/day). The enormous amount of data makes it intractable to perform manual image analysis. This analysis bottleneck is an even bigger problem in scientific imaging community with all the high throughput scattering, full-field imaging and spectroscopy facilities. To expedite material discovery research with ever-growing scientific datasets, we have to automate data analysis workflows with new data mining and machine learning algorithms.

Recently, deep learning methods [67] have achieved great success in computer
vision applications. Multi-layer “deep” learning mechanisms have proven to be of critical importance towards understanding high dimensional data and semantic concepts. In this work, we attempt to introduce deep learning techniques to resolve the aforementioned bottleneck. We take on a specific task, i.e., multi-label annotation of x-ray scattering images to study the effective “deep” practice for scientific data. We identify two major challenges of scientific datasets learning: label scarcity and pattern complexity, and articulate two key strategies: synthetic data, and physics-aware data transform — Fourier-Bessel Transform (FBT) — to address these challenges. FBT produces an energy representation of images as radial and angular frequencies and effectively represents image structure. On top of this novel transform, we build a double-view convolutional neural network (CNN) to learn features from both the original image and its FBT. We perform end-to-end inference and transfer learning experiments and demonstrate that our model consistently outperforms a standard CNN.

Our main contributions are in the following three perspectives:

• We pioneer a feasible deep Fourier-Bessel feature learning method that involves Fourier-Bessel coefficient estimation and deep CNN learning, demonstrating the effectiveness of FBT as a powerful physics-aware feature transform. To the best of our knowledge, this is the first attempt to articulate a deep Fourier-Bessel feature that is compact, informative, and discriminative.

• We articulate a Double-View Fourier-Bessel Convolutional Neural Network (DVFB-CNN) in details. Our double-view framework aims to integrate image CNN and Fourier-Bessel coefficient CNN for image annotation and exploit the power of ensemble learning.

• Our work is a proof-of-concept towards physics-awareness. Built upon our ongoing success of Fourier-Bessel learning, it is possible to enrich physics understanding during large-scale intelligent data analysis, which conduces to more success in multi-view learning with more powerful physics-aware transforms.

3.2 Related Work

X-ray Scattering Image Analysis is a topic of interest in both the x-ray and computer vision communities. [144] used spectral clustering for unsupervised clustering of images, and exploited symmetry analysis in the associated reconstruction problem [29, 111], while [49] used diffusion-based clustering. These clustering methods do not reflect semantic labels and are not suitable for multi-label problems where
3.3 Background, Challenges, and Strategies

Suppose a probing x-ray beam is shone through a sample with realspace density distribution $\rho(r)$. The resulting x-ray interference gives rise to a reciprocal-space scattering intensity of

$$I(q) = \left| \int_V \rho(r) \exp(iq \cdot r) \, dV \right|^2.$$

(3.1)
Figure 3.2: Examples of experimental images and synthetic images.

The sample’s reciprocal-space is centrosymmetric with scattering patterns centered about the direct beam. The symmetry of the scattering experiment is thus well-described by polar coordinates, \( I(q) \triangleq I(q, \phi) \), where \( q \) is the total momentum-transfer, and \( \phi \) is the azimuthal angle about the direct beam.

We now try to predict a set of non-mutually-exclusive image attributes. This is called multi-label learning as one sample can bear more than one attribute from the label set. Our label set represents characteristics of various aspects, e.g., ‘halo’ or ‘ring’ (visual), ‘isotropic’ or ‘6-fold symmetric’ (style variations), ‘powder’ or ‘polycrystalline’ (material structure). Figure 3.1 shows example images of ‘ring’ with distinct style variations.

For x-ray scattering images, and scientific data in general, there are two major difficulties with learning CNNs. The first one is label scarcity. The amount of labeled data is usually not enough to train CNN to fit. This is because scientific dataset requires domain experts to label; also it takes \( N \) times longer for an \( N \)-label dataset. The second one is pattern complexity. Unlike natural images where regions of interest are usually local, visual patterns in x-ray scattering images are often diffused and overlapping, which do not work well with the local receptive fields of CNN units. We propose two strategies to overcome these difficulties: transfer learning with synthetic data and Fourier-Bessel Transform (FBT).

3.3.1 Transfer Learning with Synthetic Dataset

We choose to train the CNN with synthetic data [131], and do transfer learning to annotate real datasets. This transfer learning scheme is inspired and proven effective by [131]. Simulation software can not only generate massive automatically labeled data, which is necessary to resolve label scarcity, but also model x-ray imagery accurately with parametric and physics-based methods [100, 114, 142], so the generated dataset is realistic and good for learning (see Figure 3.2). Various works have shown that proper simulations transfer well in deep learning [47, 72].
Figure 3.3: Fourier-Bessel basis functions on a bounded disk $r < a$ with zero-value boundary condition. The subscript $(n, m)$ refers to the radial and angular indices \ref{eqn:3.3}\ref{eqn:3.4}. Only real parts of the basis shown.

### 3.3.2 Fourier-Bessel Transform

We use Fourier-Bessel Transform (FBT) as a feature transform to better separate features. FBT is a polar coordinate based transform, which is good for exploiting image centrosymmetry; it is a Fourier-like decomposition, which is good for preserving image information and thus physics. The radial/angular basis of FBT is derived from the eigenfunctions $\Psi(r, \varphi)$ of Laplacian in polar coordinates:

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2}. \ (3.2)$$

With separation of variables $\Psi(r, \varphi) = R(r) \Phi(\varphi)$, we can solve the differential equations to compute the basis with proper boundary conditions \cite{133}.

For representing 2D images, we are particularly interested in the bounded function space, \textit{i.e.} complex functions $f(r, \varphi)$ with support $r < a$. Due to orthogonality, the normalized basis functions are:

$$\Phi_m(\varphi) = \exp(im\varphi) / \sqrt{2\pi}, \ (3.3)$$

$$R_{nm}(r) = J_m \left( x_m \cdot \frac{r}{a} \right) / \sqrt{N_n^{(m)}}, \ (3.4)$$

where $m \in \mathbb{Z}$, $n \in \mathbb{Z}^+$, $N_n^{(m)} = a^2 J_{m+1}^2(x_{mn})/2$, $J_m$ is the $m$-th order Bessel function, and \{\xmm\} are the roots of $J_m(x)$, and FBT and the inverse transform are:

$$P_{nm} = \int_0^a \int_0^{2\pi} f(r, \varphi) \Psi^*_{nm}(r, \varphi) r \, dr \, d\varphi, \ (3.5)$$

$$f(r, \varphi) = \sum_{n=1}^{\infty} \sum_{m=-\infty}^{\infty} P_{nm} \Psi_{nm}(r, \varphi). \ (3.6)$$
**ALGORITHM 2:** Fourier-Bessel Estimate.

- **Input:** Input image $I \in \mathbb{C}^{h \times w}$, precomputed basis images $\Psi \in \mathbb{C}^{B \times H \times W}$, beam center position $(x, y) \in \mathbb{R}^2$
- **Output:** Fourier-Bessel coefficients $A$ of the original image

1. Crop the basis
   
   $\Psi_c = \text{P} \Psi = \Psi (:, (H/2 - y) : (H/2 + h - y), (W/2 - x) : (W/2 + w - x))$;

2. Apply disk projection (3.8)
   
   $\Psi' = \text{P}_D \Psi_c, \ I' = \text{P}_D I$;

3. Reshape $\Psi'$ to $\mathbb{C}^{B \times hw}$, $I'$ to $\mathbb{C}^{hw}$;

4. $\tilde{\Psi}' = \begin{bmatrix} \text{Re}(\Psi') - \text{Im}(\Psi') \\ \text{Im}(\Psi') \end{bmatrix}, \ \tilde{I}' = \begin{bmatrix} \text{Re}(I') \\ \text{Im}(I') \end{bmatrix}$;

5. Solve
   
   $\min_A \| \tilde{\Psi}' A - \tilde{I}' \|_2^2 + \lambda \left[ \alpha \| \tilde{A} \|_1 + (1 - \alpha) \| \tilde{A} \|_2 \right]$;

6. $A = \tilde{A}(: B) + \tilde{A}(B:) i$;

Examples of F-B basis functions are shown in Figure 3.3.

### 3.4 Double-View Fourier-Bessel Convolutional Neural Network

In this section we describe our joint learning model, Double-View Fourier-Bessel Convolutional Neural Network (DVFB-CNN). DVFB-CNN is a network that jointly encodes both the original image and its FBT with separate convolutional sub-networks. It can be run end-to-end for training and prediction; it can also produce deep Fourier-Bessel features for generic, robust and physics-aware transfer learning (Figure 3.4).
3.4 Double-View Fourier-Bessel Convolutional Neural Network

3.4.1 Fourier-Bessel Estimate of Partial Images

As the observed image has a natural bounded support of the detector range, we model it in the function space defined on \( \{ r < a \} \) about the direct beam. However, a direct FBT is not feasible, as the image center is not necessarily aligned with direct beam position, and the disk \( r < a \) may be cropped.

Consider FBT in discretized pixel space with a cropped basis. Since the image is the inner product of F-B basis and the coefficients, we can solve a linear system to compute the coefficients. With a pixel mask applied, this can be formulated as the optimization:

\[
\min \| P \Psi A - I \|^2_F + \lambda R(A),
\]

where \( P \) is the mask, \( \Psi \) is the discrete Fourier-Bessel basis, \( A \) is the basis coefficients, \( I \) is the image, \( \lambda \) is a regularization parameter and \( R(A) \) is a regularization term.

Now given an image with center \((x, y)\), we first crop the basis to image range and apply a disk projection to eliminate the outer rings clipped by the boundary of the detector window, to alleviate boundary effect:

\[
P_D = 1_D, \quad D = \{ p \parallel p - O'\parallel < \max(x, 256 - x, y, 256 - y) \}, \tag{3.8}
\]

where \( 1_D \) is a binary mask. As for \( R(A) \), we choose the elastic net regularization to enforce sparsity and boundedness of the estimated coefficients. Following the conventional notation of sparse optimization, we vectorize the image and still denote it as \( I \), and reshape all the tensors accordingly. The optimization problem is now formulated as follows:

\[
\min \| P_D (P \Psi A - I) \|^2_F + \lambda [\alpha \| A \|_1 + (1 - \alpha) \| A \|_2], \tag{3.9}
\]

where \( \alpha \in [0, 1] \) controls the ratio of \( l_1 \) and \( l_2 \) terms. We convert problem (3.9) to a real-valued problem [116] and solve it using coordinate descent. The complete estimation algorithm is outlined in Algorithm 2.

In practice, we generate \( 256 \times 256 \) synthetic training images, set \( a = 256 \), and precompute \( 600 \times 600 \) discretized basis images \( \Psi \) to allow some beam position offset. We keep all radial frequencies up to 40 and even angular frequencies up to 20, thus \( \Psi \in \mathbb{C}^{40 \times 11 \times 600 \times 600} \). We set \( \lambda = 10^{-4}, \alpha = 0.8 \). To obtain a quick estimation, we downsample \( \Psi \) and \( I \) by half and empirically end the iterative solver after 20 iterations.

3.4.2 Double-View Convolution

By solving (3.9), we obtain the estimated Fourier-Bessel coefficients and rearrange them into a \( 40 \times 11 \times 2 \) real array. Herein, we interchangeably call the array a “Fourier-Bessel image” or “coefficient image”. To enhance the high-frequency activity, we
normalize the F-B image and perform an element-wise signed logarithmic operation:

\[ A_{\log} = \text{sign}(A) \frac{\log |1 + A|}{\log(1 + M)}, \]

where \( M = 2^8 - 1 \) is the maximum pixel intensity. We process the original image and the log F-B image with separate convolutional sub-networks: image CNN \( f(I) \) and F-B coefficient CNN \( g(A_{\log}) \). \( f(\cdot) \) and \( g(\cdot) \) are fully customizable. Here we choose AlexNet [64] for \( f(\cdot) \). For \( g(\cdot) \), we design smaller networks as F-B images are smaller. With extensive comparisons (see supplementary material), we determine a novel 4-layer network (2c-pool-2c) (Figure 3.4).

### 3.4.3 Training

**Network Output and Loss.** After obtaining the feature maps from the last convolutional layers of the double CNNs, we flatten all the feature maps to 1D and concatenate them as fused feature vectors. The fused vectors are then fed into subsequent fully-connected (fc) layers. The last layer is sigmoid output \( \sigma(x) = 1/(1 + \exp(-x)) \). Finally we minimize the binary cross entropy loss:

\[ \mathcal{L} = -\frac{1}{n} \sum_{ij} z_{ij} \log x_{ij} + (1 - z_{ij}) \log(1 - x_{ij}), \]

where \( n \) is the batch size, \( z_{ij} \) is the true binary value of attribute \( j \) of sample \( i \), and \( x_{ij} \) is the network output.

**Sub-network training and ensemble.** During training, we first learn \( f(\cdot) \) and \( g(\cdot) \) separately by feeding either one input only and optimizing the network parameters to convergence. Then we load the pre-trained weights to optimize the fc layers. We use stochastic gradient descent to train the model, with 50 images per patch and learning rate 0.1.

**Transfer Learning.** To use trained joint features to annotate experimental datasets, we take the activations of the first fc as feature vectors and apply one-vs-all classifiers to classify each label. We use RBF kernel SVMs in our experiments.

### 3.5 Experiments and Evaluations

We report experiments with two types of data: we use synthetic data to train and test DVFB-CNN, and we use fully-annotated experimental data to run transfer learning.
3.5 Experiments and Evaluations

<table>
<thead>
<tr>
<th>Feature</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-B Coefficients</td>
<td>0.3921</td>
</tr>
<tr>
<td>Log F-B Coefficients</td>
<td>0.3513</td>
</tr>
<tr>
<td>Image CNN fc (AlexNet)</td>
<td>0.6055</td>
</tr>
<tr>
<td>Coefficient CNN fc (2conv)</td>
<td><strong>0.7034</strong></td>
</tr>
<tr>
<td>Joint fc (AlexNet+2conv)</td>
<td><strong>0.7371</strong></td>
</tr>
</tbody>
</table>

Table 3.1: Linear classifier performance on feature vectors.

3.5.1 End-to-end Training and Testing: Synthetic Dataset

We use our simulation software to generate 50,000 synthetic x-ray scattering images, with 45,000 images for training and the remaining 5,000 held out for testing. We choose 10 binary labels to predict (Table 3.2), based on availability of positive samples and good visual and physical meanings for learning generic representation.

Ablation Studies. We demonstrate that both FBT and deep convolutional networks are essential for forming discriminative features — a deep Fourier-Bessel feature, by comparing the following feature vectors: (a) F-B coefficients computed from Algorithm 2; (b) Log F-B coefficients via (3.10); (c) The first fully-connected layer (fc) of a trained AlexNet; (d) The first fc of a trained two-layer F-B CNN (2conv); (e) The first fc of joint network AlexNet+2conv.

For the purpose of assessing these feature representations, we train a simple classifier: a linear SVM to classify these features with each of the 10 labels. We use the complete feature vectors, normalized to unit $l_\infty$ norm, and choose the SVM penalty factor via cross validation. We report the mean average precisions (mAPs) of all these features in Table 3.1.

The results show that FBT itself is not a discriminative representation from a classification standpoint, but deep F-B feature is, and surprisingly, even better than deep image convolutional features. Moreover, joint fc feature confirms the effectiveness of double-view ensemble learning.

Performance. We choose AlexNet as image CNN, and 2c-pool-2c as F-B CNN, to train and evaluate the joint model end-to-end. We report the average precisions (APs) per attribute in Table 3.2. To understand the APs with imbalanced attributes, we also list the ratios of positive samples per attribute, which equal to the APs of random classifiers, as a low baseline.

Comparing AlexNet and F-B CNN, the latter shows significantly better performance for symmetry related attributes, which demonstrates that FBT is closely related to the physics of experiments and effective to encode structures, and thus well complements image CNN features. The joint network features successfully take advantage of the two and improve the performance of most single labels, resulting in a
Table 3.2: Average precision per label (synthetic dataset).

![Image](image_url)

Figure 3.5: Masked Fourier-Bessel Estimate. (a) Original image. (b) Mask. (c) Recovered image. (d) Estimated coefficients. Magnitude of complex numbers shown.

13% increase in mAP from AlexNet in image space.

3.5.2 Transfer Learning: Experimental Dataset

We collected a wide range of x-ray scattering images from different experiments and beamlines to comprehensively reflect the heterogeneity and diversity of real experimental data. The majority of images are transmission small-angle x-ray scattering (TSAXS) or wide-angle scattering (WAXS), with a small fraction of images from grazing-incidence (GISAXS) measurements. The data were collected from the experiments conducted at the X9 beamline of the National Synchrotron Light Source (NSLS) and the CHX and CMS beamlines at NSLS-II and all fully annotated by a
3.5 Experiments and Evaluations

Figure 3.6: Average precisions on experimental dataset. The two figures share the same legend.

<table>
<thead>
<tr>
<th>Mask/Ctr. Position</th>
<th>Single</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td># Labels</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td># Images</td>
<td>2,429</td>
<td>2,718</td>
</tr>
<tr>
<td>Train-Test Split</td>
<td>2,000 / 429</td>
<td>2,300 / 418</td>
</tr>
</tbody>
</table>

Table 3.3: Experimental data setup.

domain expert.

To assess the generalizability of the joint deep feature, we undertake experiments on two groups of data: single-experiment and mixed. The single dataset only contains the CHX images of NSLS-II under the same experimental setting, while the mixed dataset contains images from multiple experiment runs on different beamlines. The mixed dataset has more positive samples of many attributes and thus presents more style differences (and deviations) from one particular run and synthetic dataset. All images are resized to $256 \times 256$. To determine a prediction label set, we pick the labels with at least around 100 positive samples. These include coarse labels (e.g., ‘Higher Orders’) and fine-grained ones (e.g., ‘Ring: Isotropic’); some of them are quite imbalanced. In total, 12 labels are chosen for single dataset, and 20 for mixed. Table 3.3 shows the experimental configuration for these two groups of data.

Masked Fourier-Bessel Estimate. X-ray scattering experiments usually have gaps and beamstops in the images, which causes some obstructed pixels, as shown in Figure 3.5b. We replace the window projection $\mathbf{P}$ in (3.9) with a masked projection...
Double-View Image Feature with Fourier-Bessel Transform

### Table 3.4: Comparison with state-of-the-art methods on experimental dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Single mAP</th>
<th>Mixed mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>lhpphog</td>
<td>0.5968</td>
<td>0.5581</td>
</tr>
<tr>
<td>PCANet</td>
<td>0.6660</td>
<td>0.5951</td>
</tr>
<tr>
<td>AlexNet fc</td>
<td>0.8189</td>
<td>0.7768</td>
</tr>
<tr>
<td>VGG-16 fc</td>
<td>0.8312</td>
<td><strong>0.7997</strong></td>
</tr>
<tr>
<td>ResNet-50 fc</td>
<td>0.8231</td>
<td>0.7084</td>
</tr>
<tr>
<td>Joint fc</td>
<td><strong>0.8513</strong></td>
<td><strong>0.7989</strong></td>
</tr>
</tbody>
</table>

Table 3.4: Comparison with state-of-the-art methods on experimental dataset.

as follows:

\[ \mathbf{P}_m = \mathbf{1}_m \mathbf{P}, \]  

where \( \mathbf{1}_m \) is a mask determined by the experiment. Figure 3.5 shows the process of the masked Fourier-Bessel Estimate.

**Performance and Comparisons.** We choose AlexNet as image CNN, and 2c-pool-2c as F-B CNN, to extract deep F-B feature descriptor from the first joint fc layer. We compare this descriptor with a few feature vectors, especially some from deeper CNNs: (a) lhpphog descriptor \[61\]; (b) PCANet \[12\]; (c) The first fc of AlexNet \[64\]; (d) The first fc of VGG-16 \[118\]; (e) The first fc of 50-layer ResNet \[38, 131\].

We use RBF kernel SVMs to classify each attribute. The SVMs are trained on full feature vectors, normalized to unit \( l_{\infty} \) norm. SVM parameters \( C \) and \( \gamma \) are determined via grid search for each attribute. We report the mAPs of all the feature vectors in Table 3.4 and show the APs and positive ratios per attribute in Figure 3.6. Results show that our joint feature consistently outperforms image based CNN that were just “deeper”; it is also much faster to train than larger networks such as VGG-16 or ResNet-50.

### 3.6 Chapter Summary

In this chapter, we have proposed a double-view learning model, DVFB-CNN, for x-ray scattering image annotation. We introduced Fourier-Bessel transform to extract feature coefficients and devised a novel coefficient CNN to learn structural information from images.

Our experiments have proven that FBT is effective to encode physical structures of x-ray images that spatial convolutions fail to capture. This deep feature transform approach easily outperformed CNNs that were simply deeper. We chose FBT as the backbone of our method not only from the observation of symmetric patterns caused by CDI, but also due to its inherent connection to the imaging process as
a Fourier transform. Thus, FBT is expected to produce structured representations, and we devised the Fourier-Bessel estimate algorithm to compute FBT for unaligned, partially obstructed x-ray scattering images. Our sparsity based estimation algorithm effectively acts as an image completion step that repairs image defects, demonstrating that a few F-B coefficients do describe the images well.

Our DVFB-CNN also demonstrated a unified, extensible ensemble learning framework design. Ensemble, as a powerful tactic in machine learning, can be utilized in deep neural networks by designing proper “fuse” layers to combine multiple features. In our model, the concatenation of the dual CNN features proves to be simple yet effective. For future applications, it is natural to exploit a few input modalities and transformations to produce different features, and then concatenate them to form a more comprehensive feature representation. This is be an easily-adoptable method to incorporate physics-aware machine learning and domain-specific machine learning in other fields.
Chapter 4

Image Feature Enhancement with Attentional Aggregation

In this chapter, we address a key difficulty in recognizing x-ray scattering images — the important patterns are weak and scattered, and heavily interfered by background noises. We present attentional aggregation as a two-step strategy to enhance image features. First, attention acts locally to highlight physically meaningful local regions; then, aggregation condense them into non-local representations to better represent their implication of physics meaning as a group. They work together to generate more recognizable features for annotation.

Figure 4.1: Examples of x-ray scattering images. The original image is a single-channel intensity map captured on the detector; it is shown here with false color for visualization purposes. We briefly describe the challenging features and the affected attributes below the images: (a) related peaks are scattered and far apart; (b) signal too dim; (c) thin vertical bars (Bragg rods) overlaying bright regions (Yoneda).
4.1 Introduction

In recent years, deep learning and convolutional neural networks (CNNs) have moved on from mostly general successes in standard computer vision applications to working with more specialized data and problems, e.g. applications in scientific communities. These dedicated applications usually come with a relatively small dataset and unique challenges with data distributions. This causes a lot of trouble for CNNs to capture them properly and realize their full potential. From a feature space’s point of view, visual features in scientific imaging lie in a low-dimensional, highly restricted and “narrow” subspace compared to natural images, and the feature representations must be enhanced properly to work with CNNs.

We will illustrate this difficulty of feature space with examples of x-ray scattering images, which are the subject of our analysis in this chapter. In x-ray scattering, a beam of x-rays is directed through a material sample of interest, diffracted by the ordering within the sample; the far-field pattern of scattered rays is captured on an x-ray area detector. This diffractive imaging process is essentially described by a Fourier transform of the sample’s real space distribution $\rho(r)$:

\[
I(q) = \left| \int \rho(r) \exp(iq \cdot r) \, dV \right|^2,
\]

from which only the intensity is captured, while the phase is lost. Thus, the diffraction is non-invertible. To circumvent the inverse reconstruction problem, material scientists can directly inspect the image to deduce a set of characteristics, e.g. visual appearance (‘halo’ or ‘ring’), style variations (‘isotropic’ or ‘6-fold symmetric’), material type (‘powder’ or ‘polycrystalline’) or crystal lattice structure (‘BCC’ or ‘lamellar’). In other words, this is an multi-label image annotation problem in the reciprocal space (or q-space); multi-label meaning the aforementioned attributes are non-mutually-exclusive.

When we apply CNNs to this multi-label annotation problem, the key challenge comes from the weak and scattered patterns in the x-ray scattering images. Here we show a few difficult cases in Figure 4.1. In Figure 4.1(a), a sample is bearing the attribute “polycrystalline”, implied by the set of high-intensity peaks in the image. The peaks themselves are local, yet the “polycrystalline” character can only be inferred by identifying a non-local set of peaks combined. If a CNN unit is to perceive a window large enough to capture all the peaks, it will take in a much bigger portion of background and possibly other overlapping signals, overwhelming the peaks in question. Figure 4.1(b)(c) shows two other cases of difficult attributes caused by weak visual features and overlapping. For weak, noisy features like these, we need effective measures to encode them robustly.
In this chapter, we present the Attentional Aggregation Module (AAM), a modularized two-step strategy to enhance feature representations. Given a certain set of convolutional feature maps, first we attempt to explicitly reweight and highlight key features with attention mechanism. When generating attention maps, we decompose them into channel and spatial attention components for better separation between them. In the spatial attention component, we design a mechanism to generate multiple spatial attention maps to apply to partitions of the feature maps, which diversifies the attentional features for different attributes of interest. Then, we attempt to condense those scattered sparse features by feature aggregation. We extend the classic Bag-of-Words (BoW) with learnable parameters so that it can be performed in-network. We apply AAM multiple times in the network at different depths so that the feature enhancement is multi-scale.

Our main contributions in this paper are as follows:

- We designed the Attentional Aggregation Module using differentiable layers and learnable parameters, which enabled end-to-end forward and backward flows in the CNN, and repeated deployment in multiple CNN layers. Thus, feature enhancement is seamless and multi-scale;

- We improved the attention modules with new tactics and multiple spatial attention maps to specifically tackle the multi-label learning problem, and demonstrated their benefits via experiments.

### 4.2 Related Work

**X-ray scattering image analysis.** Studying x-ray scattering imagery is an interdisciplinary effort of computer vision and scattering communities. There are unsupervised methods such as spectral clustering [144] and diffusion-based clustering [49], as well as supervised methods such as [61] using handcrafted image descriptors. CNN based techniques are first used in other similar scientific dataset problems, e.g. [143] applies a CNN to classify x-ray protein crystallization images. For the x-ray scattering image annotation problem, [97] performs 1D convolutions on the circular average curve of the images; [131] implements residual learning [38] and convolutional autoencoders; [35] proposes a joint learning framework with physics-aware feature transform [133]. Despite the general success of these methods, some attributes are connected to more intricate features as we explained in Figure 4.1, and thus they are hard to observe by even humans and so remain difficult for machine learning methods.

**Feature aggregation.** Typical feature aggregation methods organize generic features by encoding statistics of a collection of features, e.g., Bag-of-Words (BoW) [120],
Figure 4.2: Left: Architecture of our network with Attentional Aggregation Modules. Right: Structure of the aggregation module.

VLAD [3, 54], Fisher Vector [101, 102], spatial pyramid matching [66] and Bag-of-Feature-Graphs [44].

Deep CNN generates dense collections of features. Many works attempt to incorporate classic feature aggregation methods, e.g. MOP-CNN [31] pools VLADs of multi-scale CNN activations, and NetVLAD [2] presents a learnable VLAD in deep CNNs. Later [87] generalized the learnable construction of NetVLAD to BoW and Fisher Vector. On the other hand, since multi-scale is naturally implied in the depths of CNNs, researchers have tried countless network designs to fuse cross-layer features as a form of feature aggregation, e.g. U-Net [108] reusing mid-layer feature maps. More sophisticated connection designs include recombinator networks [41] and stacked hourglass networks [90], and [77] has made some detailed discussions about various pathway designs in multi-scale analysis. However, many of these methods rely on reasonably good local features to aggregate. They mostly work like representing the composition of a scene given all the objects have been well depicted. As for x-ray scattering images, the features are tricky to capture even at the local level and we need explicit strategies to boost local features so that aggregation can be effective.

Attention mechanism. It is known that humans perceive images not by observing the entire scene, but focusing their attention on salient regions [65]. In computer vision, researchers have attempted to mimic attention for feature learning [65] and generative models [122, 139]. For computing attention maps, researchers have proposed to use fully-connected MLP [139], convolutions and residual convolutions [132], and correlations to encode non-local interactions [147]. With the recent success of channel attention in SENet [45], dual attention, which is a decomposition of channel and spatial attention, has become a popular method to model attention [13, 78, 135].

4.3 Attentional Aggregation Module

In this section we describe the Attentional Aggregation Module (AAM), which is designed as learnable network layers and applied several times to enhance the features in a multi-scale fashion. The structure of AAM is described in Algorithm 3, and the overall network is shown in Figure 4.2.
ALGORITHM 3: Attentional Aggregation Module.

**Input:** Convolutional feature \( F \in \mathbb{R}^{C \times H \times W} \).

**Output:** Attentional feature \( F_{\text{attn}} \in \mathbb{R}^{C \times H \times W} \) and aggregated vector \( f_{\text{aggr}} \in \mathbb{R}^{K} \).

1. Compute channelwise attention \( M_{C}(F), F' \) using (4.3), (4.7);
2. Compute spatial attention maps \( \{M_{S}(F')\}_{i} \) using (4.8);
3. Split \( F' \) along the channel dimension into \( p \) slices \( \{F'_i\}_i \);
4. Compute spatial attention \( M_{S}(F') = \text{Concat}(\{M_{S}(F') \otimes F'_i\}_i) \), and \( F'' \) using (4.3);
5. Compute attentional features \( F_{\text{attn}} \) using (4.6);
6. Reshape \( F_{\text{attn}} \) to \( \bar{F}_{\text{attn}} \in HW \times C \);
7. Compute \( A = \bar{a}(\bar{F}_{\text{attn}}) \in HW \times K \) using (4.11);
8. Sum \( A \) up along the columns and normalize to unit 2-norm to get \( f_{\text{aggr}} \);

**Figure 4.3:** Structure of CBAM [135].

**Figure 4.4:** Our channel and spatial attention modules.
4.3.1 Attention: Local Refinement

Given a feature map $F \in \mathbb{R}^{C \times H \times W}$, we first attempt to refine the features locally using attention mechanism. Attention in features is essentially a reweighting process to highlight certain parts of the features, represented by multiplying $F$ with attention map $M$. When the attention module is trained to compute from the data $F$ to mimic human attention, it can be written as

$$F' = M(F) \odot F,$$

where $\odot$ represents elementwise multiplication, and $M(\cdot)$ becomes a data-driven attention estimator.

**Dual attention.** We use a dual attention to approximate $M$, *i.e.* decompose the overall attention into 2 multiplicative components: 1D channel attention and 2D spatial attention

$$F' = M_C(F) \odot F, \quad F'' = M_S(F') \odot F'.$$

We can think of $M_C$ as modulating the $C$ feature channels, or amplifying/suppressing the $C$ feature detectors if we consider each channel as a specific detector; and $M_S$ focuses on pixel locations. This is similar to a low-rank matrix decomposition and the attention in different dimensions can be better separated.

We follow a typical dual attention model — Convolutional Block Attention Module [135] (CBAM, shown in Figure 4.3) — to formulate $M_C$ and $M_S$. The channel attention component first spatially pools $F$ to a $C$-dimension vector, and then encodes the vector with a 2-layer fully-connected network (or Multi-Layer Perceptron, MLP):

$$M_C(\cdot) = \sigma \circ \text{MLP} \circ \text{Pool}_C(\cdot),$$

where $\sigma$ is sigmoid. Similarly, the spatial attention component first performs a channelwise pooling to generate a $H \times W$ matrix, and then computes a convolution:

$$M_S(\cdot) = \sigma \circ \text{Conv} \circ \text{Pool}_S(\cdot).$$

Finally, we add up the reweighted $F''$ with $F$ and normalize it with batch normalization [52] (BN) to prevent feature degradation due to successive multiplications with values between $[0, 1]$ [132]:

$$F_{\text{attn}} = \text{BN}(F + F'').$$

Unfortunately, the original CBAM in the network does not improve the annotation on our x-ray scattering datasets, because smaller dataset size and fewer positive samples cause more difficulty to learn. We made a few improvements for the attention mechanism, as follows:
4.3 Attentional Aggregation Module

Pre-activation normalization of attention maps. We find that both channel and spatial attention components tend to saturate sigmoid and generate attention maps that are all 0 or 1. To correct this, we add a BN prior to sigmoid, in both channel and spatial components, to stabilize the range of features. The channel attention component is now written as

\[ M_C(\cdot) = \sigma \circ BN \circ MLP \circ Pool_C(\cdot). \quad (4.7) \]

With normalization, the generated attention maps can actually have values in \([0, 1]\).

Multiple spatial attention maps. We argue that for spatial attention, one single spatial map to reweight all \(C\) feature channels does not account for the different features that these channels specialize in for multi-label annotation. Instead, we feed the feature \(F'\) into \(p\) duplicate branches of the spatial attention component:

\[ \{ M_i^S = \sigma \circ BN^i \circ Conv^i \circ Pool^S | 1 \leq i \leq p \}. \quad (4.8) \]

Then, we split \(F'\) into \(p\) uniform slices along the channel dimension \(C\), and reweight each \(C/p\)-channel slice \(F'_i\) with \(M_i^S(F')\). Then we concatenate all the slices into a \(p\)-way reweighted \(F''\). In our experiments, we set \(p = 4\).

Specialized loss for multi-maps. We further push the \(p\) spatial attention branches to diversify. For this purpose, we partition the attributes into \(p\) groups and associate each group with one spatial attention branch. For each branch during training, we compute the label loss with respect to its own attribute group, and update its parameters with this specialized loss/gradient only.

For instance, we denote the image attributes as \(Y = \{y_1, y_2, \ldots, y_n\}\). We may associate \(Y^{(1)} = \{y_1, \ldots, yn/p\}\) with \(M_1^S(\cdot)\), \(Y^{(2)} = \{yn/p+1, \ldots, 2n/p\}\) with \(M_2^S(\cdot)\), and so forth. Pick a label loss function, e.g., binary cross entropy:

\[ L_{BCE}^\Lambda = -\frac{1}{|\Lambda|} \sum_j \Lambda_j \log o_j + (1 - \Lambda_j) \log(1 - o_j), \quad (4.9) \]

where \(\Lambda\) is a certain attribute set, \(y_j\) is the true value of attribute \(j\), and \(o_j\) is the prediction value. For training the rest of the network, our objective is to fit all the attribute predictions, \(\Lambda = Y\); For \(M_\Lambda^S(\cdot)\), \(\Lambda = Y^{(i)}\).

The full attention module is shown in Figure 4.4.

4.3.2 Aggregation: Non-local Representation

We then aggregate the local attention feature \(F_{\text{attn}}\) to condense non-local feature representations. Our feature aggregation module is based on the classic Bag-of-Words (BoW). Formally, given a corpus of sampled feature vectors (words), BoW computes
a clustering to determine a set of $K$ clusters $\{C_k\}$ and their centroids $\{c_k\}$. For an image with $N$ extracted features $\{f_i\}$, its BoW is its feature distribution with respect to the clusters

$$\left[ \sum_{i=1}^{N} a_1(f_i), \sum_{i=1}^{N} a_2(f_i), \ldots, \sum_{i=1}^{N} a_K(f_i) \right],$$

(4.10)

where $a_k(\cdot)$ is an assignment function, typically a hard 0-1 assignment determined by the nearest $c_k$.

Consider $F_{\text{attn}}$ as an $H \times W$ pool of $C$-dimensional feature words and we attempt to aggregate them spatially. In order to perform aggregation in-network and have it compatible with back-propagation, the key is differentiable formulation for the cluster assignment $a_k(\cdot)$. NetVLAD [2] manages to do this by replacing the hard assignment $a_k(\cdot)$ with soft assignment and relaxing the learnable parameters. Learnable BoW is similar, as we need to replace $a_k(\cdot)$ with a softmax

$$\tilde{a}(x) = \text{softmax}(\{w_k^T x + b_k\}_{1 \leq k \leq K}),$$

(4.11)

and then all the $K$-assignment vectors are summed up and normalized, denoted as $f_{\text{aggr}}$. In our experiments, we set $K = 64$. The aggregation module is shown in Figure 4.2 on the right.

### 4.3.3 Network Architecture

We adopt VGG-16 [118] as our backbone network. VGG-16 is naturally divided into 5 blocks with pooling layers in between, implying different scales. We put in residual bypass [38] over each block to separate the features at each scale. After the last convolution, we feed the feature maps into a global average pooling layer and a sigmoid output.

We plug in an AAM before the addition in each of the aforementioned residual bypasses. The AAM computes $F_{\text{attn}}$ and $f_{\text{aggr}}$. We pass $F_{\text{attn}}$ through onto subsequent CNN layers, and $f_{\text{aggr}}$ leads to a fully-connected layer and a sigmoid, which serves as a side output $y_s, 1 \leq s \leq 5$. We fit the side output to ground truth attributes, similar to [121]. The purpose is to stimulate multi-scale features to better relate to attributes. As a result, the overall loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}^\Lambda(y, o) + w \sum_{s=1}^{5} \mathcal{L}_{\text{BCE}}^\Lambda(y_s, o),$$

(4.12)

where $\Lambda$ is a specialized attribute set (described in Section 4.3.1), $y$ is the output from the last CNN layer, $y_s$ is a side output, and $o$ is the real attribute. We set $w = 0.2$. 
4.4 Experiments

4.4.1 Datasets and Metrics

We use the following 3 datasets to evaluate our network:

**Synthetic Dataset.** We use simulation software [141] to generate high volumes of simulated x-ray scattering images with auto-generated attributes. The software models x-ray imagery with high fidelity [100, 114, 142] and adapts well for machine learning models that extend to real data [35, 131]. For comparisons with previously reported methods, we generate 45,000 images for training and 5,000 images for testing as our synthetic dataset. We pick 20 attributes with typical visual appearances and/or physical meanings to predict.

**Experimental Dataset.** We take the experimental dataset assembled in [35] to assess our method with real experimental data. The experimental dataset is collected from various x-ray beamline facilities and fully annotated by a domain expert. It is organized into 2 groups: “single” consists of different image captures with homogeneous experiment setups (beam center position, detector placement etc.), and “mixed” where experiment setups are diverse. The single dataset has 2,000 training images, 429 testing images and 12 attributes; and the mixed dataset has 2,300 training images, 418 testing images and 20 attributes. These attributes are not the same as those in the synthetic dataset.

**Fashion-MNIST.** To test the AAM with other forms of data, we choose Fashion-MNIST [137] as a general purpose dataset. It is designed as a drop-in replacement of the heavily-used MNIST [69] and consists of grayscale images of 10 types of clothing articles of size $28 \times 28$, 60,000 training samples and 10,000 testing samples.

For synthetic and experimental datasets, the annotation is multi-label, and we report the average precisions (APs) per attribute and mean average precision (mAP); Fashion-MNIST is a multi-class classification dataset, and we report the classification accuracy.

4.4.2 Ablation Studies

To verify the effect of the attention and aggregation modules, we trained and tested the CNN on synthetic dataset, with our proposed strategies added incrementally. They are listed in Table 4.1 as: (a) barebone VGG-16, (b) with aggregation module, (c) with aggregation and normalized CBAM, (f) with AAM, but without specialized loss, and (g) with AAM. Since CBAM has the saturation problem (described in Section 4.3.1), we added a BN in (c).

We trained all the 5 networks end-to-end and directly evaluated the outputs. For specialized loss, the 20 attributes were grouped as follows: (1) major visual
elements: Diff low-q, Diff hi-q, Halo, Higher ord, and Ring; (2) symmetry: Sym halo, Sym ring, 2-fold, 4-fold, and 6-fold; (3) texture: Anisotropic, Isotropic, Spotted, and Textured; (4) visual style: Orientation: sharp, Orientation: broad, Orientation: interm, Width: sharp, Width: broad and Width: interm. We report the APs per attribute in Figure 4.5, and the mAPs in Table 4.1.

We can see consistent improvements from (a)(b)(c)(f)(g) with the added components:

- The performance gain from (a) to (b) comes from feature aggregation and early side supervisions, which effectively shape the multi-scale features as proven by GoogLeNet [121];

- (c)(f)(g) shows the benefit of multiple spatial attention maps, and demonstrates that specialized loss provides the additional information to effectively train more learnable parameters and operations.
4.4 Experiments

![Comparison of transfer learning performance using different deep learning methods, on experimental dataset.](image)

**Figure 4.6:** Comparison of transfer learning performance using different deep learning methods, on experimental dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Single</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>0.8312</td>
<td>0.7997</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>0.8231</td>
<td>0.7084</td>
</tr>
<tr>
<td>DVFB-CNN</td>
<td>0.8513</td>
<td>0.7989</td>
</tr>
<tr>
<td>SENet</td>
<td>0.8723</td>
<td>0.8071</td>
</tr>
<tr>
<td>Residual Attention</td>
<td>0.8837</td>
<td>0.8183</td>
</tr>
<tr>
<td>Ours</td>
<td>0.8739</td>
<td>0.8225</td>
</tr>
</tbody>
</table>

**Table 4.2:** Comparison with state-of-the-art deep learning methods, on experimental dataset.

4.4.3 Comparison with Other Attention Methods

We also compared AAM with some other attention mechanisms: (d) SENet [45] and (e) residual attention module [132], as shown in Table 4.1. We kept the VGG-16 with AAM structure unchanged but swapped all the attention modules with the other methods.

The results show that AAM enables better precision. Essentially, SENet is channel attention only, and residual attention does not decompose the attention into channel and spatial components, while AAM exploits both dual attention and residual attention, and also utilizes multiple spatial attention maps.
4.4.4 Transfer Learning with Experimental Dataset

We used the experimental dataset to evaluate transfer learning. The reason to perform transfer learning is that real experimental data is not enough for training a CNN. This is a common hurdle for many specific applications and datasets. For example, our experimental dataset has less than 3,000 images. Therefore, to train on a bigger set of synthetic data is a crucial strategy to actually use the CNN.

We input the images into the trained network from Section 4.4.2 and computed the global average pooling layer as feature vectors. Then we normalized them and used them to predict the images’ attribute with RBF kernel SVMs. SVM parameters $C$ and $\gamma$ were determined via cross validation.

We compared our method with some state-of-the-art deep learning methods: VGG-16 [118], ResNet-50 [38] and DVFB-CNN [35], as well as the other attention methods trained in Section 4.4.3: SENet and residual attention. We followed the same experiment setup as in [35] and compared with the APs reported therein, shown in Figure 4.6. We also list the mAPs in Table 4.2.

We can conclude from the results that our proposed feature enhancement improves the features further than much deeper networks like ResNet-50; it is even better than DVFB-CNN without precomputed feature transforms or assumption of structural symmetry, and thus our method is more general. AAM shows comparable results among the attention methods which consistently improves the annotations. In particular, AAM has the best mAP in the mixed dataset, showing that multiple spatial attention maps are capable of handling discrepancies of different experimental and imaging setups.

4.4.5 Classification with Fashion-MNIST

To show AAM is applicable to other tasks and network configurations, we ran classification experiments on Fashion-MNIST. We set up 3 CNNs of different depths to learn to classify the 10 types of clothing articles. The layer configurations and prediction accuracy are shown in Table 4.3. Experiments show improvements in accuracy in all of the setups, and thus prove that AAM is equally effective to enhance the features for general datasets.

4.4.6 Attention Visualization

For qualitative assessment of the attentional features, we computed Grad-CAM [113] to visualize the attribute related activities in our network. We show in Figure 4.7, from left to right, the input image, and Grad-CAM visualizations of the last convolutional layers, in the networks without attention (VGG-16 + Aggr.), with Normalized
4.5 Chapter Summary

In this chapter, we detailed a multi-label visual feature learning framework with the Attentional Aggregation Module (AAM). We validated with the experiments that these modules served the purpose of enhancing image features from local to global, which is crucial to understand scientific images with weak, scattered and noisy features. The representations of local and non-local features are interconnected, because feature aggregation methods rely on precomputed local features. Thus, we designed AAM to tackle the local refinement and non-local representation simultaneously. Our data-driven attention mechanism enhances weak features that are known to be challenging in x-ray scattering data, with innovative channel-spatial decomposition and multiple spatial attention designs, and its effect is confirmed by both performance gain and verification by visualizations. Our learnable aggregation module and side output effectively shape the features on a non-local level. Experiments showed that

<table>
<thead>
<tr>
<th></th>
<th>2-layer</th>
<th>3-layer</th>
<th>5-layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv-64</td>
<td>conv-64</td>
<td>conv-64</td>
<td>conv-128</td>
</tr>
<tr>
<td>(AAM)</td>
<td>maxpool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conv-128</td>
<td>conv-128</td>
<td>conv-256</td>
<td></td>
</tr>
<tr>
<td>conv-256</td>
<td>conv-256</td>
<td>conv-256</td>
<td></td>
</tr>
<tr>
<td>(AAM)</td>
<td>maxpool</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Avg Pool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fc-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>softmax</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>0.9050</td>
<td>0.9166</td>
<td>0.9294</td>
</tr>
<tr>
<td>+AAM</td>
<td><strong>0.9202</strong></td>
<td><strong>0.9269</strong></td>
<td><strong>0.9393</strong></td>
</tr>
</tbody>
</table>

Table 4.3: CNNs to predict Fashion-MNIST and prediction accuracy.

We can see the activation regions continue to improve with more sophisticated attention setups. Take (b) as an example: Normalized CBAM and AAM both correctly identify the 2 disjoint high intensity areas that implied 2-fold symmetry, and the Grad-CAM activity of AAM is more precise. This shows that our multiple spatial attention maps can indeed adapt to diverse attributes and react to different features accordingly.
they significantly improved the final classification result, even though the side outputs were not directly used in the evaluation stage.

Our network with AAM showcases a network design to enhance image features in a multi-scale fashion due to the multiple AAMs in different depths of the CNN. Also, residual connections separate the features of different scales, and all differentiable operations in AAM guarantees the network is trained and evaluated end-to-end. We showed that multi-scale analysis can be formulated with minimal modifications to a deep neural network with the introduction of a differentiable network block that can be deployed repeatedly.

Figure 4.7: Grad-CAM visualization of x-ray scattering image samples.
Chapter 5

Ptychography Phase Retrieval

In this chapter, we present PtychoNet, a convolutional encoder-decoder network to perform ptychographic reconstruction, also known as “phase retrieval”. Our network offers a non-iterative approach to reconstruct ptychography by reversing the diffraction with deep learning, and dramatically improves the reconstruction under poor overlap conditions, which calls for a rethinking for the previously known overlap conditions necessary for a successful reconstruction.

5.1 Introduction

Coherent diffractive imaging (CDI) is a technique of measuring the scattering from an object illuminated by a coherent beam, e.g. laser light or X-ray. For CDI, the analysis of the object relies on a reconstruction algorithm to recover the object structure based on the measured diffraction pattern. However, the reconstruction problem is generally ill-posed, because the intensity of a diffraction pattern is captured on the detector while the phase information is lost. Phase retrieval methods are thus developed to recover both the amplitude and phase of the object of interest [98].

Among the CDI methods, ptychography offers the advantage of imaging large objects with high resolution. The method captures multiple diffraction patterns of an object with a set of shifted localized illuminations [24, 106]. The overlap of the illuminations provides redundancy for robust reconstruction of the object with a resolution that is not limited by the probe size nor the scanning step size but the highest detectable scattering angle. X-ray ptychography allows for the reconstruction of object structures on the nanometer scale in 3D [19], practical for studies in e.g. biology, geology, and material science. Electron ptychography provides sub-angstrom resolution for 2D materials [56] and optical ptychography allows the imaging of large and thick samples with 3D isotropic micron-resolution [75, 85].
Challenges of ptychography reside in its long data acquisition time and the computational complexity in data processing. The scanning nature of the method is advantageous for imaging extended field of views of an object, however, at the cost of long experimental time and the increased likelihood of serious radiation damage of the sample. For X-ray imaging, it is predictable that with synchrotron upgrades, the coherent photon flux will increase by a few orders of magnitude, presenting a great computational challenge to reconstruction algorithms if corresponding increase in acquisition speed is also achieved with upgraded instrumentation. Therefore, sparse scanning schemes and fast reconstruction algorithms will bring immediate benefits to experiments, including high-throughput and low-dose imaging, as well as provide a solution to the elevated computational demand in the near future. In this work, we introduce a new ptychographic reconstruction method based on machine learning, presenting a possible direction to overcome the challenges.

5.1.1 Formulation

In CDI, an object of interest is shone by a light source and the incident wave is diffracted by object structure. On the far-field detector, this results in a Fourier transform

\[ I = |\mathcal{F}(\psi)|^2, \]  

(5.1)

where \( \psi \) is the exit wave of the object. In ptychography, the illumination (“probe”) is shifted laterally to generate multiple views of the object:

\[ I_j(q) = |\mathcal{F}[P(r - r_j)O(r)]|^2, \]  

(5.2)

where \( q \) is the coordinate in the reciprocal space, \( r \) is in real space, \( r_j \) is the probe position, \( P \) is the probe and \( O \) is the object. The reconstruction problem, from diffraction images to real space, is called “phase retrieval” because the far-field detector only captures the intensity, given by Eq. (5.1), and the reconstruction is essentially recovering the phase. Traditional methods of reconstruction are based on iteration to satisfy the real space and Fourier constraints alternately [25]. These methods normally take a few hundred iterations to converge.

We presented PtychoNet, a deep learning based method to perform phase retrieval for ptychography. PtychoNet reverses the diffractions at each scanning point, and then merges them to generate the amplitude and phase of the original object. Using PtychoNet, we are able to compute phase retrieval in only one forward pass without any iterations.

5.1.2 Contributions

Our contributions in this chapter are mainly as follows:
5.2 Related Work

- We describe the first non-iterative, end-to-end method to solve the phase retrieval problem for real-space ptychography without limitation on scanning patterns.
- We presented a successful demonstration of PtychoNet through simulations and show its robustness under extreme low overlap conditions when conventional methods failed.
- By using PtychoNet output to initialize iterative algorithms, we show that reconstruction of fine features is made possible for a dataset with almost no overlap, which is not achievable previously. This new finding urges the re-examination of the ptychography overlap conditions and expands the potential of ptychography.

5.2 Related Work

**Generative deep learning.** Convolutional neural networks (CNNs) [64, 68] for image synthesis have seen remarkable progress among recent successful applications. Early methods such as variational autoencoder [63] have some success but often suffer from blurry images. Auto-regressive models like PixelCNN [110, 128, 129] compute each pixel progressively using a probability model, which is efficient but struggles to generate larger images. In order to generate sharp and detailed images, researchers use convolution transpose ("deconvolution") to build deep neural networks that enable high-resolution upsampling, notably in image segmentation [82, 93]. Generative Adversarial Networks (GANs) [32, 53, 103] train a pair of generator network and discriminator network as adversaries and drastically improve the quality of generated samples.

**Ptychography and related techniques.** Ptychography [24, 106] is a lensless coherent diffractive imaging method that utilizes redundancy for high resolution. For 3D objects, ptychographic X-ray computed tomography (PXCT) offers quantitative electron density and absorption tomograms with nanoscopic resolution [18, 19, 40]. Electron ptychography [50] and optical ptychography [73, 85] also provide high-resolution in their respective imaging regimes. On the other hand, Fourier ptychography microscopy (FPM) [148] generates high-resolution image by combining a collection of images with different illumination angles. Fourier ptychographic tomography (FPT) [42] at the optical regime shows reconstruction of 3D object at sub-micron resolution.

**Ptychographic reconstruction methods.** Conventional reconstruction methods for ptychography are based on iterative phase retrieval algorithms [25], *e.g.* PIE [24, 107], ePIE [81], difference map (DM) [123, 124] and maximum likelihood [36, 125]. Recently automatic differentiation is employed to solve phase retrieval with more
straightforward formulations and better parallelization support \cite{28, 55, 89}. There are also closed form formulations using Wigner-distribution deconvolution (WDD) \cite{6}, but WDD is highly susceptible to noise. Recently, techniques such as iterative refinement \cite{74} and low-rank matrix completion \cite{71} are adopted to suppress the noise, achieving comparable results with aforementioned well-established methods. 

**Machine learning reconstruction methods.** Machine learning is revolutionizing reconstruction algorithms. Paine and Fienup propose to compute smart initial guesses for phase retrieval using CNNs \cite{96}. At present, there is a lack of machine learning applications in the standard real-space ptychography, but there have been demonstrations of machine learning in FPM. PtychNet \cite{58} and cGAN-FP \cite{9} use CNNs to reconstruct FPM on simulation datasets, and \cite{91, 92, 140} extend CNN-based methods to sequential FPM and test on real experimental data. In general, CNN generative methods work well with a wide range of imaging setups, e.g. optical microscopy \cite{104}, limited-angle tomography \cite{33}, holography \cite{105}, imaging through scattering media \cite{76}, nonlinear electromagnetic inverse scattering \cite{73} and imaging of phase-only objects \cite{119}. However to our knowledge, there has been no demonstration of ptychographic reconstruction based on diffraction patterns using machine learning.

### 5.3 Notation

We first describe the ptychographic imaging and reconstruction in discretized pixel space. We define the object $O$ as a complex-valued $H \times W$ image, and the size of probe $P$ is $h \times w$. With $P$ fixed, the full ptychographic scan is determined by a set of $N$ probe positions, e.g. rectangular mesh pattern, concentric pattern or Fermat spiral pattern \cite{51}. For convenience to describe algorithms, we can store all the scanning positions by recording the top, bottom, left, and right pixels $[t, b, l, r]$ in a layout matrix $M \in \mathbb{Z}^{N \times 4}$. With an abuse of notation, we define the projection operator:

$$M_j(O) = O[M[j, 0] : M[j, 1], M[j, 2] : M[j, 3]], \quad (5.3)$$

to acquire the valid range of the diffraction image at scanning position $j$, and \cite{52} becomes

$$I_j = \frac{1}{hw} ||\text{DFT}([P \cdot M_j(O)])||^2, \quad (5.4)$$

where $1/hw$ is a normalization factor for Discrete Fourier Transform (DFT). What the detector actually captures is the phase-less intensity $I_j \in \mathbb{R}_{+}^{h \times w}$. For phase retrieval, we directly use the amplitude $A_j = \sqrt{I_j}$ instead of $I_j$. We stack up all $A_j$ in a new dimension to form $A(O; P, M) = [A_1, A_2, \ldots, A_N] \in \mathbb{R}_{+}^{N \times h \times w}$ and define it as a ptychography or a “full scan” of the input object $O$ under probe $P$ and layout $M$. 
5.4 PtychoNet

PtychoNet is a convolutional encoder-decoder network to compute real space images from ptychographic scans (architecture shown in Figure 5.1). In order to solve such an inverse problem, we try to exploit every image and meanwhile enforce consistency in overlapping regions. For that purpose, our strategy is to reconstruct every single one of the images, and then stitch the local patches later according to their spatial relations.

The input of the network is a full scan $A$ with known layout $M$. The encoder side encodes each diffraction image $A_j = A[j,:,:]$ individually and the decoder side computes their corresponding object patches $Y_j$. Here we represent the patches using their amplitude and phase, $Y_j \in \mathbb{R}^{2\times H \times W}$. For the final output, we initialize an all-zero output matrix $Y \in \mathbb{R}^{2\times H \times W}$ and a counter matrix $K \in \mathbb{Z}^{2\times H \times W}$. We add each $Y_j$ to $Y$ and increase the counter $K$ on the its scanning position. Finally we average the sums of all the output patches to compute the full object. The complete reconstruction is described in Algorithm 4.

**ALGORITHM 4:** Reconstruction using PtychoNet.

| Input: | Full scan $A \in \mathbb{R}^{N\times h \times w}$, scan layout $M \in \mathbb{Z}^{N\times 4}$. |
| Output: | Object image $Y \in \mathbb{R}^{2\times H \times W}$. |

1. $Y = K = 0^{2\times H \times W}$;
2. for each diffraction image $A_j$ in parallel do
3.   Compute the corresponding object patch $Y_j$ in real space with input $A_j$;
4.   $M_j(Y) = M_j(Y) + Y_j$;
5.   $M_j(K) = M_j(K) + 1$;
6. end
7. $Y = Y / \max(K, 1)$;
Figure 5.2: Probe and scan layout. (a) probe amplitude; (b) probe phase; (c) layout of Fermat spiral pattern. Green dots show the centers of every scanning position and red rectangle shows the size of the probe w.r.t. object.

For the loss to minimize during training, it is feasible to include terms such as pixelwise loss, perceptual loss [20] and adversarial loss [32]. Here we propose two losses. The first is a plain mean square error (MSE)

$$\mathcal{L}_1(Y, O) = \mathcal{L}_{MSE}(Y, O) = \frac{1}{HW} \| Y - O \|^2.$$  

(5.5)

Apart from measuring the difference in the real space, we can also constrain the reciprocal space. Imagine we perform ptychography on the output $Y$ instead of the real object, we shall also obtain a ptychographic scan pattern that is close to the input $A$. Using (5.4), we can compute the MSE of computed and true diffractions, simply called DFT loss

$$\mathcal{L}_{DFT}(Y, O; A, P, M) = \frac{1}{Nhw} \sum_{j=0}^{N} \left\| \frac{1}{\sqrt{hw}} \| \text{DFT}[P \cdot M_j(Y)] \| - A_j \right\|^2,$$  

(5.6)

and thus we propose the dual space error (DSE)

$$\mathcal{L}_2(Y, O; A, P, M) = \mathcal{L}_{MSE} + \lambda \mathcal{L}_{DFT}.$$  

(5.7)

In our experiment, $\lambda = 1$.

5.5 Experiments

5.5.1 Implementation Details

We generated a simulation dataset for training using Caltech-256 Object Category Dataset [34]. To construct a real space “object”, we paired two images from the dataset, one as amplitude map and one as phase. The amplitude was scaled and shifted to $[0.5, 1]$ and the phase was in $[-\pi/3, 0]$. We split these images into two
disjoint sets, from which we generated 2,000 training objects and 100 test objects respectively. We generated a fixed probe of size $128 \times 128$ and performed ptychography scans on all images using Fermat spiral pattern [51], shown in Figure 5.2. In our resolution settings, our images were resized to $478 \times 480$ to cover the scanning trajectory and 354 diffraction images were computed during a complete scan. We used Adam optimizer [62] to train PtychoNet with a learning rate of 0.0002 and $\beta_1 = 0.5$.

5.5.2 Comparison with Iterative Algorithms

We compared the reconstruction of PtychoNet with difference map (DM) method. For PtychoNet, we trained the network using MSE (5.5) and DSE (5.7) separately. For DM, the initial object $O_g$ was random, and the initial probe was the actual probe $P_g = P$. 300 iterations were applied. Reconstruction results are shown in Table 5.1. The probe is handled differently in PtychoNet and in iterative algorithms. In our dataset, the probe was fixed and it was directly provided to DM as initial value. This is crucial for the algorithm to converge, as the probe is a common factor in all the diffractions. It is advised to provide a reasonable approximation of the probe as well as a spatial support of the probe to ensure that the iterative algorithm converges to an optimal solution. In PtychoNet with MSE, however, the probe was not explicitly given in either training or testing and it was learned from data. There was no explicit formulation of a matrix $P$ and PtychoNet learned to directly reverse the diffraction patterns in Fourier space to the real space object.

On the other hand, DSE utilized the known probe $P$. Comparing Table 5.1(b) and (c), we can see that the DFT loss in (c) corrected the contrast in some of the images, because MSE alone tends to push the pixel output to a mean value, causing the image to look gray and bland overall, and Fourier constraint alleviated this problem. The parameter $\lambda$ can be tuned to adjust this correction.

5.5.3 Tests on BrainMaps Image Dataset

We also demonstrate here some success in transferring the model learned on natural images to optical micrograph of biological samples. Biological specimen often presents hierarchical structures and thus a broad spatial spectrum. Using PtychoNet trained with Caltech-256 (Section 5.5.1), we computed phase retrieval on BrainMaps [1], a biological microscopic image set. We selected 50 full images and 50 cropped image patches (see Table 5.2(a)) from Dataset 107 [1] to generate 100 real space objects. For each object, one image was used for phase, scaled and shifted to $[0.5, 1]$, and amplitude was set to constant $0.5$. The same probe from Section 5.5.1 was used to perform

\[\text{http://brainmaps.org/index.php?action=viewslides&datid=107}\]
Table 5.1: Training and reconstruction on Caltech-256. (a) Ground truth. (b) PtychoNet (MSE). (c) PtychoNet (DSE). (d) Difference map.
5.5 Experiments

Table 5.2: Reconstruction on BrainMaps dataset using a Caltech-256 trained PtychoNet. (a) Ground truth. (b) PtychoNet (MSE). (c) PtychoNet (DSE).

ptychography. The results are shown in Table 5.2. We can see that even though the contrast and fine details were not perfect, because it was trained on a different dataset, PtychoNet was still able to capture the structure of the images. This cross-modality result shows the great potential for modeling more imaging setups and materials of interest.

5.5.4 Performance with Low Overlap

Experimentally the current bottleneck of ptychography is often the data acquisition time and the amount of dose deposited on sample. Sparse sampling with useful resolution can dramatically increase the imaging throughput as well as enable imaging of radiation-sensitive materials or the study of in-situ dynamics. Here we compare the results of PtychoNet and DM with known probe when the overlap constraint was severely weakened by a sparse scan. We reduced a 400-point mesh pattern to 25 points, shown in Figure 5.3. We plotted all the scanning positions and a few probe windows to show the overlap size. All small light green and big dark green dots together give the 400 center positions for scanning points. With this dense scan, the adjacent points are 17 pixels apart, giving an 87% overlap between adjacent probe windows. Using all 400 points gives a fairly good overlap, so we can expect to have a clean reconstruction similar to Table 5.1. The sparse case is only using the 25 big dark green dots. This reduces the overlap to 34%. Considering the probe is almost zero outside the central disk, which is essentially the spatial extent of the illumination, the overlap is estimated to be around 60% in diame-
Table 5.3: Comparison of reconstructions with low overlap using Caltech-256. (a) Ground truth. (b) PtychoNet (MSE). (c) DM with known probe.
5.6 Extreme Low Overlap Reconstruction

PtychoNet is a non-iterative method for phase retrieval where the output is produced with deterministic number of operations, different from an iterative approach where hundreds of iterations are taken before a chosen stopping criterion is met. However, PtychoNet as a deep neural network essentially acts as a deep approximator learned from data. Even though it is capable of producing plausible, close-to-real results, it cannot be quantitatively accurate for all samples and it is prone to shifts in data distributions when working with new data. Iterative algorithms, on the other hand, are good at refining the output steadily under proper conditions.

For iterative algorithms, a good initialization is especially crucial for data with low overlap. Here we demonstrate a case where using PtychoNet as a good initial guess for iterative algorithms is actually necessary to reconstruct fine features. In

Table 5.4: Comparison of reconstructions with low overlap using Caltech-256. (a) DM, object initialized randomly. (b) PtychoNet output. (c) DM, object initialized by PtychoNet.

Figure 5.4: Comparison of iterative descent using PtychoNet and random initializations.

We can see the iterative algorithm failed to converge because there was barely any overlap to constrain the object $O$, but PtychoNet managed to learn the inverse problem from data and still produced a reasonable estimate. Such dramatic reduction in the scanning sampling, a factor of 1/16 in this initial demonstration, shows that PtychoNet could be a powerful method to overcome the aforementioned bottleneck.
the low overlap scenario shown in Section 5.6, both PtychoNet and DM reconstructions suffered from varying degrees of degradation. We compare two initializations of the object using the DM method: one is random, and the other is the PtychoNet (MSE) output. The test object is taken from Table 5.3(a) (0, 0). Three hundred iterations were applied and the reconstructions are shown in Table 5.4. We report the MSE and the error measured by the normalized update size \( \chi(t) = \sqrt{\sum \|\mathbf{O}(t) - \mathbf{O}(t-1)\|^2} / \sqrt{\sum \|\mathbf{O}(t)\|^2} \) in Figure 5.4. We can see that PtychoNet provided a good reconstruction of low spatial frequency components and thus, with the PtychoNet initialization, the starting error was significantly lower than that of random initialization. The final reconstruction (c) was much better than what PtychoNet and DM alone could achieve. This shows that under adverse conditions (low overlap), machine learning with iterative refinement is a great combination for dramatic enhancement in reconstruction quality. Based on these results, it can be concluded that PtychoNet by itself and PtychoNet-DM essentially appeals for re-examination of the overlap requirement for ptychography as well as offers great potential for studies to be performed in regimes previously deemed impossible due to the limited temporal and spatial resolution achievable in experiments.

5.7 Chapter Summary

In this chapter we presented PtychoNet, an non-iterative end-to-end method for ptychographic reconstruction. We have demonstrated successful reconstructions using the method and presented cross-modality result, from the model trained with Caltech-256 and transferred to biological samples, giving confidence in the broad applicability of PtychoNet. We have also shown that the number of scanning positions can be reduced by over an order of magnitude while the reconstruction still approximates the ground truth. On the contrary, using DM by itself with this sparse scanning pattern gives reconstructions with almost no recognizable features. Moreover, when using PtychoNet result as an initial guess for iterative algorithms, high spatial frequency components and fine features are successfully reconstructed, which to our knowledge was not achieved before for dataset with such a sparse scanning. This result may lead to the re-assessment of the optimal overlap and true potential of ptychography.
Chapter 6

Other Work and Conclusions

In this chapter, we introduce our ongoing work on interpretation of ptychography reconstruction models, flexible ptychography reconstruction, and fluorescent image super-resolution. Also we briefly introduce another application on small dataset feature learning using PDE-guided diffusion.

6.1 Ongoing Work

6.1.1 Interpretation of PtychoNet

In Chapter 5 we presented PtychoNet, a generative neural network to perform reconstruction for ptychography. We know that the probe $P$ is not explicitly modeled in the network, but implied; we may think that PtychoNet learns to reverse CDI and divide the exit-wave by $P$ at the same time. However, this does not help to make the physics process transparent within the network. We are interested in how neural networks understands the experiment setup in ptychography. Specifically, can we extract physics parameters from the learned neural networks?

Our simulation software is able to generate different probes given a few parameters. For instance, we can adjust the propagation distance $d$, which is the distance between the material sample and the focal point of the lens, to change the probe. In our

![Figure 6.1: Architecture of probe autoencoder.](image-url)

61
(a) Ground truth, phase of 16 probes.  
(b) Autoencoder output.

**Figure 6.2:** Output of probe autoencoder.

---

**Figure 6.3:** Estimating the propagation distance $d$ of 100 randomly generated probes, compared to ground truth.
Figure 6.4: Two different setups of flexible ptychography reconstruction. (a) Input images $I$, generate object $O$ and probe $P$: $[O, P] = f(I)$; (b) Input images $I$ and probe $P$, generate object $O$: $O = f(I, P)$.

Experiments, $d$ is subject to uniform distribution between 50 and 60 microns. Now we train an autoencoder to reconstruct the probe, and we attempt to estimate $d$ from the probe encoder. From the middle feature layer, we add a logistic regressor to fit to $d$. The loss function is expressed as

$$
\mathcal{L} = \frac{1}{hw} \| y_P - P \|^2 + \frac{1}{\mu} | y_d - d |^2,
$$

where $h, w$ are the side lengths of the probe, $y_P$ is the autoencoder output, $\mu$ is the mean propagation distance, and $y_d$ is the regressor output. The architecture of the autoencoder is shown in Figure 6.1. The reconstruction result is shown in Figure 6.2 and the regressor estimate of propagation distance is shown in Figure 6.3. We can see that the probe parameters can be reliably estimated from the probe encoder.

We also consider to identify the probe from trained PtychoNet. Sellam et al. proposed DeepBase [112], a system to inspect neural network behaviors. They introduce the so-called hypothesis functions to model a certain behavior, e.g. looking for a specific word, and evaluate the correlations between the neurons to examine and the hypotheses. Following their method, we can define hypothesis functions to match the probe images, and inspect the CNN units throughout the training process. We can compare the results with a few different probes and see if there is any specific pattern of correlated neurons, and thus understand how the probe is represented in the generative network. With this knowledge, we will be able to separate the probe in the phase retrieval, and even replace the probe robustly in a trained PtychoNet.

### 6.1.2 Flexible Ptychography Reconstruction

PtychoNet has one major limitation — it assumes the probe $P$ stays constant in the entire training set as well as testing set. For different experiment setups that have different $P$, the current version of PtychoNet would need to be retrained with a new training set using the same $P$, which is inefficient and cumbersome. For more flexible
ptychography reconstruction, we attempt to design a network that is able to perform clear reconstruction for varying probes.

Based on the encoder-decoder network structure, two different modifications can be quickly applied to accommodate the varying probe. The first method is to keep the encoder from PtychoNet and produce two outputs on the decoder (generator) side, one being the object $O$ and the other $P$. However, since the phase retrieval problem is already difficult in the fixed probe case, adding the probe problem to it severely disrupts the training. As a result, the network fails to produce meaningful outputs. The second method is to provide both the diffraction images $I$ and the probe $P$ as network inputs, extract features from them both, and compute the object $O$ as final output. This setup is also more similar to the conventional iterative algorithms where the probe $P$ needs to be known in practice. The two network setups are shown in Figure 6.4.

In theory, the dual encoder setup provides all the necessary information for the reconstruction, but in our experiments the output is still much more blurry compared to fixed probe PtychoNet. We attempted two strategies: first, we trained the entire network from scratch; second, we pretrained the image encoder, probe encoder and object decoder using autoencoders respectively. In both cases, the image quality does not have noticeable difference. We hypothesize the reason is the probe encoder does not provide sufficient information about the probe in the reconstruction, even when we confirm the pretrained autoencoder can reconstruct the probe. To verify this, we compared the results of dual encoder with a network that has a single image encoder only. This is essentially a fixed probe PtychoNet applied to a varying probe dataset. The result is shown in Figure 6.5 — no visible improvements from (b) to (c).

Figure 6.5: Comparison of reconstructed phases of fixed and varying probe ptychography.
6.1 Ongoing Work

Figure 6.6: Training the reconstruction network with latent space features. With a fixed probe, the image features go straight to a standard feature and the network acts as the original PtychoNet. When the probe changes, the network is trained to warp the new features to the standard feature as if we were using one same probe all the time.

To handle this issue with ineffective probe encoder, we propose to manipulate the latent space with a two-step training scheme that place training tasks in the middle feature encoding layers. In this modified network, the diffraction image features and probe features are concatenated in the middle, and they would go through a few additional layers before moving on to the object decoder, as shown in Figure 6.6. The training is split into two stages. First, we train the network without the probe encoder, using diffraction images from a fixed probe, as if it were a standard PtychoNet until the reconstruction is good. After the first stage finishes, the network is able to compute a latent encoding of any object input in the middle feature layer. They correspond to the same fixed probe, and we consider them the “target” feature, to which we will warp features from new probes. In the second stage, we fix the image encoder and object decoder and train the probe encoder and middle convolutional layers only, using diffraction images generated from a series of varying probes. The training objective is to fit the latent feature encoding to the “target” feature under the probe used in the first stage. In other words, using the probe encoder and the additional transformation layers, we map the features back to a “standard” known probe.

6.1.3 Fluorescent Image Super-resolution

Multimodality imaging allows for more efficient and thorough inspection of materials. In the beamline HXN at National Synchrotron Light Source II (NSLS-II), Brookhaven National Laboratory (BNL), simultaneous acquisition of ptychographic data and fluorescence data is routinely employed to obtain both the electron density map and elemental maps, respectively. Ptychography has the advantage of achieving a resolution that is not limited by the probe size and the scanning step size, while for fluorescence the imaging resolution is often limited by these factors. To achieve high-resolution multimodality imaging, a small probe has to be used, which would
cause the acquisition time to increase (on the order of hours or days for one sample). However, the fact that ptychography can offer a high-resolution map of the sample structure even with a large probe and sparse scanning leads us to explore the possibility of overcoming the probe size limitation on the fluorescence imaging (FI) resolution. The ability to reconstruct in high resolution the electron density map and elemental maps from a sparse data set (or large probe) can potentially dramatically increase the data acquisition rate, and as a result, increase the sample throughput as well as enhance the temporal imaging resolution. Thus, it is desirable to have a super-resolution model to enlarge a set of fluorescent images given a single high-resolution reference image. In this case, the fluorescent image super-resolution (FISR) problem is a mixture of super-resolution and image translation, as the source image and target image are different in appearances and not in the same domain.

SRGAN is an adversarial learning super-resolution framework proposed by Letig et al. [70]. They presented a pair of generator $G$ and discriminator $D$ to solve the adversarial min-max problem:

$$
\min_G \max_D E_{I_{\text{HR}} \sim p_{\text{train}}(I_{\text{HR}})}[\log D(I_{\text{HR}})] + E_{I_{\text{LR}} \sim p_{G}(I_{\text{LR}})}[\log(1 - D(G(I_{\text{LR}})))],
$$

(6.2)

where $I_{\text{LR}}$ is a low-resolution image and $I_{\text{HR}}$ is a high-resolution image. They defined a perceptual loss function for training SRGAN, consisting of three components. The first component is a pixel-wise MSE

$$
l_{\text{SR \text{MSE}}} = \frac{1}{Z_1} \sum_{x,y} (I_{\text{HR}} - G(I_{\text{LR}}))_x^2,
$$

(6.3)

the second component is a VGG loss, measuring the Euclidean distance between the feature representations of $G(I_{\text{LR}})$ and $I_{\text{HR}}$ in the VGG-19 network [118]

$$
l_{\text{SR \text{VGG/i,j}}} = \frac{1}{Z_2} \sum_{x,y} (\phi_{i,j}(I_{\text{HR}}) - \phi_{i,j}(G(I_{\text{LR}})))_x^2,
$$

(6.4)

where $\phi_{i,j}$ represents the post-activation features at the $j$-th convolution between the $(i - 1)$-th and $i$-th maxpooling layer in VGG-19, and the last component is an adversarial loss

$$
l_{\text{SR Gen}} = \sum \log D(G(I_{\text{LR}})).
$$

(6.5)

The overall perceptual loss function is expressed as

$$
l_{\text{SR}} = l_{\text{SR \text{MSE}}} + l_{\text{SR \text{VGG/i,j}}} + 10^{-3} \cdot l_{\text{SR Gen}}.
$$

(6.6)

On the other hand, Consider the FISR problem using the classic deconvolution approach for image deblurring. A low-resolution fluorescent image is generated by a
clean image convolved with a kernel $K$

$$I_{fi} = K * I,$$  \hspace{1cm} (6.7)

where $I$ is a clean image. Theoretically, $I$ can be solved by a deconvolution given the known $K$. In FI, $K$ can be estimated, but it is usually not precise, and thus a direct deconvolution is not ideal. This can be expressed as

$$I_{fi} = (K_0 + n_0) * I + n_1,$$ \hspace{1cm} (6.8)

where $n_0$ is some uncertainty to the estimated $K_0$, and $n_1$ is some noise in the image.

We propose to compute FISR with a generative neural network. Similar to Section 6.1.2, we are faced with a varying $K$ problem — training the FISR network with a fixed $K_0$ is more stable, but it is inconvenient for application. If the network can take a varying kernel,

$$K = K_0 + n_0,$$ \hspace{1cm} (6.9)

as a parameter, then a single trained network can solve different FISRs robustly. We propose to use a two-stage training scheme similar to the varying probe problem (Section 6.1.2). The FISR network consists of a low-resolution (LR) encoder, a kernel ($K$) encoder and a super-resolution (SR) decoder. First, we train the LR encoder and SR decoder with a fixed $K_0$. Then, we train the $K$ encoder with a set of LR images generated by varying $K$, warping the new $K$ induced features to the fixed $K_0$.

6.2 Small Dataset Feature Learning Using PDE-guided Diffusion

In this section, we present a human-oriented feature detection framework, learning diffusion on global graph (LDGG) to understand personalized interests in a simple and lost-cost way. LDGG is a new approach to detect and recommend regions of personalized interest (RoPIs) on 3D models (meshes), given a small learning set that has RoPIs specified by the user. In general, a region of personalized interest is a subset of mesh vertices or faces that has a certain functional or semantic meaning. It is specific to the interest of individual human users or the recognition task, and any two RoPIs are not necessarily able to establish very dense vertex-to-vertex correspondences, nor partially overlap with each other up to some rigid body motion. For example, they can be the legs among a few 3D meshes of different animals, or the wings in both birds and aeroplanes, and they do not look alike at all. We construct a global graph to represent the training and testing dataset and formulate the feature detection problem based on a learned anisotropic diffusion guided by PDEs, avoiding
the need for large volumes of training data that are typically necessary for machine learning models. The overall framework is shown in Figure 6.7.

**Global Graph Construction.** First we describe in detail how the global graph is represented and constructed. Instead of constructing the global graph on all the vertices on the meshes, we partition them into “supervertices” to serve as vertices in the graph $G = (V, E)$ to reduce the number of graph vertices while reflecting local geometry. Specifically, we perform normalized cuts (NCuts) \[30\] where each vertex is associated with the concatenation of its normal and shape diameter function (SDF) \[115\]. The number of supervertices each mesh is empirically set to 200 and it can be adjusted when needed.

The user is asked to specify RoPIs on mesh vertices by determining a vertex as the central point as well as the interest scope. The interest scope is measured by the biharmonic distance field \[79\] about the central point. The biharmonic distance between the $i$-th and $j$-th vertices is defined as

$$D_{bh}(i, j)^2 = \sum_{k=1}^{m} \left( \frac{\chi_k(i) - \chi_k(j)}{\gamma_k} \right)^2,$$

where $\gamma_k$ and $\chi_k(\cdot)$ are the first $m$ eigenvalues and their corresponding eigenfunctions of the Laplacian-Beltrami operator on the mesh. It forms a set of equidistance contours about the central point, and the user determines an interest scope, within which all vertices are marked as RoPI.

The edge set $E \subset V \times V$ is constructed with considerations of both geometry and personalized interests. First, we model the feature information on mesh vertex/face level by computing the following set of feature descriptors: Shape Diameter
6.2 Small Dataset Feature Learning Using PDE-guided Diffusion

Function (SDF), Gaussian Curvature (GC) [26], Conformal Factor (CF) [8], Shape Context (SC) [7], Average Geodesic Distance (AGD) [39], and Wave Kernel Signature (WKS) [5]. For each supervertex, we gather all these 6 descriptors at the vertices/faces within and compute a histogram for the value distributions for each descriptor. Finally, we concatenate the histograms to construct the feature vector for each supervertex, denoted as \( \{h_p, p \in \mathcal{V}\} \).

We follow [80] to compute the affinity matrix \( A \) for all the feature vectors \( \{h_p\} \) in the high-dimensional feature space. First we put all \( \{h_p\} \) as column vectors in a feature matrix \( X \) and compute the singular value decomposition (SVD) of \( X \):

\[
X = USQ, \quad (6.11)
\]

and we obtain the feature-driven graph affinity matrix of \( G \) as

\[
A = QQ^T. \quad (6.12)
\]

We then trim the graph according to personalized interests and needs for diffusion as follows:

\[
A(i, j) = \begin{cases} A(i, j), & (i, j) \in \mathcal{E}, \\ 0, & (i, j) \notin \mathcal{E}. \end{cases} \quad (6.13)
\]

Three types of edges are in the set \( \mathcal{E} \): (1) \( p_i \) and \( p_j \) are neighbors on a testing mesh; (2) one of \( p_i \) and \( p_j \) is on a training mesh, the other on a testing mesh, and \( A(i, j) \) is in the \( m \) largest weights in \( A(i, \cdot) \); (3) \( p_i \) is in a RoPI and \( A(i, j) \) is in the \( m \) largest weights in \( A(i, \cdot) \). \( m \) is empirically set to 5.

**Diffusion on Global Graph.** The RoPI detection problem can be formulated as interest diffusion on \( G \). We define a real-value interest score function \( f : \mathcal{V} \rightarrow \mathbb{R} \), which is analogous to temperature in heat diffusion. We denote the seeds of interest as \( \mathcal{S} \), or heat source, and let \( f(p) = 1, p \in \mathcal{S} \) as source temperatures. Let \( e \) be an environment node with zero temperature, and all the vertex \( p \) are connected to \( e \). Mathematically, the result of interest diffusion is described by a linear elliptic system with Dirichlet boundary (LESD):

\[
F(f, \nabla f) = 0, \quad (6.14)
\]

\[
f(e) = 0, \quad (6.15)
\]

\[
f(p) = 1, p \in \mathcal{S}, \quad (6.16)
\]

where \( \nabla f = [f(p) - f(q_1), \ldots, f(p) - f(q_{|N_p|})] \), \( N_p = \{q_1, \ldots, e\} \) is the neighborhood set of \( p \). \( F \) can be any smooth function and we use an anisotropic diffusion term to control the diffusivity:

\[
F(f, \nabla f) = \text{div}(K \nabla f), \quad (6.17)
\]
where $K$ is an inhomogeneous metric tensor

$$K(p) = \text{diag}(A(p, q_1), \ldots, A(p, q_{|N_p|-1}), z_e),$$ (6.18)

where $z_e$ is a small constant, for measuring the dissipation conductance at $p$.

For the purpose of propagation the interest scores from the seeds to all nodes in $V$, we can define the objective as maximizing the sum of scores while the diffusion should be stable:

$$\max_{S \in \mathcal{M}^n} T(S) = \sum_{p \in V} f(p; S),$$ (6.19)

$$\text{s.t. } (6.14) - (6.16) \text{ holds},$$

where $\mathcal{M}^n = \{S|S \subset V, |S| \leq n\}$. As demonstrated in [81], the optimal seed set $S$ is not the RoPIs $\tilde{S}$ on the training meshes, even though it is intuitive. The stable interest scores along with the optimal seed set are solved by submodular optimization.

**Diffusion Guidance for Learnable PDEs.** To make the diffusion both geometry- and interest-aware, we introduce a guidance to our PDE-directed diffusion system. We add a fidelity term to the governing function $F$:

$$F_u(f, \nabla f) = \text{div}(K \nabla f) + \lambda P_{V\setminus S}(f - f_u),$$ (6.20)

where $\lambda \geq 0$ is a parameter, $f_u$ is a guidance map, and $P_{V\setminus S}$ is a projection on $V \setminus S$. The PDE formulation can be discretized as

$$f(p) = \begin{cases} \frac{1}{d_p + \lambda} \left( \sum_{q \in N_p} K(p, q)f(q) + \lambda f_u \right), & p \in V \setminus S, \\ 1, & p \in S, \end{cases}$$ (6.21)

where $K(p, q)$ is the element on the diagonal of $K(p)$ corresponding to node $q$ and $d_p = \sum_{q \in N_p} K(p, q)$.

For computing $f_u$, we model the diffusion guidance to consider the most and least similar regions with RoPIs with foreground and background guidance. We compute the foreground guidance $f_F$ using (6.21) without the $f_u$ term while setting $\tilde{S}$ as source seeds. For background guidance, we empirically pick the top 70% supervertices that are farthest away from each supervertex in RoPI in the high-dimensional feature space, denoted as $B_i$, and take the intersection $B = \bigcap B_i$. We use (6.21) again to compute the non-interest score $f_B$ and let $f_B = 1 - f_B$. The final guidance map $f_u(p) = f_F(p) + f_B(p)$. 
At last we introduce yet one more penalty term to suppress the redundancy in $\tilde{S}$:

\[
U(S) = \sum_{p \in S} u(p),
\]

(6.22)

\[
u(p) = \exp(-f_u(p)^2),
\]

(6.23)

and the complete LDGG framework is formulated as the following combinatorial optimization:

\[
\max_{S \in M^n} T(S) - U(S),
\]

(6.24)

s.t. \( F_u(f, \nabla f) = 0 \),

\[
f(e) = 0,
\]

\[
f(p) = 1, p \in S.
\]

**Experiment Results.** For evaluating the LDGG framework, we assemble the data from 19 categories in the Princeton Segmentation Benchmark (PSB) [15], 20 meshes each, and 4 categories from Sidi et al. [17], including Lamp (20 meshes), Candelabara (28 meshes), Iron (18 meshes), and Guitar (44 meshes). We use the accuracy measure defined in Kalogerakis et al. [57]:

\[
\text{Acc} = \frac{\sum_i a_i I(y_i, y^*_i) + 1}{\sum_i a_i},
\]

(6.25)
Other Work and Conclusions

Figure 6.9: Experiment results of RoPI detection using LDGG. The red dashed circles highlight the refinements enabled by the additional training mesh on the left.

where $a_i$ is the area of the $i$-th face, $y_i$ is the ground truth label for the $i$-th face, $y^*_i$ is the estimated label, and $I(y_i, y^*_i)$ is an indicator function that equals to 1 when $y_i = y^*_i$ and $-1$ otherwise.

We demonstrate the effect of the number of training meshes on detection accuracy and show that LDGG is capable of learning with a small learning set. Among 10 categories, we randomly pick 1-4 meshes as training meshes to perform RoPI detection, and show the accuracy scores in Figure 6.8. We can conclude from the scores that three training meshes are sufficient for accurate and semantically sound RoPI detection results. A qualitative inspection in shown in Figure 6.9. We can see as training meshes are added, the ambiguity in functional parts is greatly reduced, and the LDGG method correctly identifies semantically similar parts, even when the source and target meshes are not structurally similar, nor do they have the same number or general positions of those parts. Still, 3 training meshes are a rather small number in the dataset, which is completely different from any conventional machine learning method that typically requires far more training data than testing data.

We also show that LDGG is robust when there are imperfections in user interactions or geometry defects. In Figure 6.10, we show 3 cases where the user ‘over-select’ or ‘under-select’ the RoPI, or the training mesh has bad geometry, like noises or holes. RoPI detection works correctly and respects the functional meaning in the testing meshes in all 3 cases.
6.3 Thesis Proposal Summary

In this proposal, we presented our recent research and ongoing work involving analysis and reconstruction of coherent diffractive imaging. Our methods covered both discriminative (classification) problems and generative (reconstruction) problems and specifically address the challenges posed by dedicated scientific datasets. In particular, we incorporated domain knowledge in the process of extracting features and designing algorithms and made our proposed systems physics-aware.

Our main contributions include:

- We presented a novel deep Fourier-Bessel feature to extract structural information from x-ray scattering images and devised a double-view convolutional neural network with this feature for classification.

- We devised a multi-scale feature enhancement method to improve image features locally and globally.

- We developed a fast non-iterative ptychography reconstruction method that solved the classic phase problem with deep learning, and showed that the learned model had the potential of relaxing the existing overlapping condition well known in the ptychography community.

All our novel methods discussed here are deeply rooted in the physics understanding, and their formulations all evolve around the nature of diffractive imaging and the underlying Fourier transform. Information such as symmetry, diffused non-local
features and the ptychographic geometry all originate from the image formation we understand, and our extensive experiments have proven we utilized them well to serve the purpose of physics aware learning.

Our future work will focus on two directions. First, we focus on improving our understanding and interpretation of existing deep learning models, in particular our proposed methods, to reveal the physics connections in the data flows of neural network so that we can reuse the successful designs in future applications. On the other hand, we plan to develop more generalized deep learning models that work with a wider range of input and more challenging data. For example, the varying probe problem in Ptychography and the varying kernel problem in FISR both seek to find a parametric solution to a difficult inverse problem with the help of deep learning. We will continue to make interdisciplinary efforts to develop robust machine learning applications that truly benefit the scientific community.
Publications

• **Guan, Ziqiao**, Kevin G. Yager, Dantong Yu and Hong Qin. “Multi-Label Visual Feature Learning with Attentional Aggregation.” *In submission.*


• Meister, Nicole, **Ziqiao Guan**, Jinzhen Wang, Ronald Lashley, Jiliang Liu, Julien Lhermitte, Kevin Yager, Hong Qin, Bo Sun, and Dantong Yu. “Robust and scalable deep learning for X-ray synchrotron image analysis.” In *2017 New York Scientific Data Summit (NYSDS)*. IEEE, 2017.

Bibliography


