Physics-assisted machine learning for X-ray imaging

by

Zhen Guo

B.A., University of California, Berkeley (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2022

Author .....
Department of Electrical Engineering and Computer Science
January 20, 2022

Certified by .....
George Barbastathis
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Physics-assisted machine learning for X-ray imaging

by

Zhen Guo

Submitted to the Department of Electrical Engineering and Computer Science
on January 20, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

X-ray imaging is capable of imaging the interior of objects in two and three dimensions non-invasively, with applications in biomedical imaging, materials study, electronic inspection, and other fields. The reconstruction process can be an ill-conditioned inverse problem, requiring regularization to obtain satisfactory reconstructions. Recently, deep learning has been adopted for 2D and 3D reconstruction. Unlike iterative algorithms which require a distribution that is known *a priori*, deep reconstruction networks can learn a prior distribution through sampling the statistical properties of the training distributions. In this thesis, we develop a physics-assisted machine learning algorithm, a two-step algorithm for 2D and 3D reconstruction. The 2D case is studied in the context of randomized probe imaging to retrieve quantitative phase distribution using deep k-learning framework, and 3D case is under X-ray tomography to retrieve the structure of integrated circuit via physics-assisted generative network. In contrast to previous efforts, our physics-assisted machine learning algorithm utilizes iterative approximants derived from the physical measurements to regularize the reconstruction with both known physical prior and the learned priors. The advantages of using learned priors from machine learning in X-ray imaging may further enable low-photon nanoscale imaging. Note that part of this thesis has been previously reported [1, 2].

Thesis Supervisor: George Barbastathis
Title: Professor of Mechanical Engineering

Acknowledgments

I would like to thank my supervisor – Prof. George Barbastathis for his invaluable supervision, support and tutelage during the course of my master degree. I would like to express gratitude to Mo Deng and Iksung Kang for their helpful discussions, in particular Abe Levitan for his treasured support which was really influential in shaping our collaboration and critiquing my numerical results. Note that Chapter 2 to 5 in this thesis are based on the paper I co-authored with Abe Levitan. I would also thank Zachary Levine, Bradley Alpert, and Michael Glinsky for their efforts in conducting numerical analysis of X-ray tomography. Note that Chapter 6 to 9 in this thesis are based on the paper I co-authored with Jung Ki Song, Michael E. Glinsky, Courtenay T. Vaughan, Kurt W. Larson, Bradley K. Alpert, and Zachary H. Levine. I would acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this thesis. Finally, my appreciation goes to my family and friends for their encouragement and support all through my studies.

Contents

List of Figures

List of Tables

Chapter 1

Single-frame 2D imaging

Diffractive imaging is a set of lensless imaging techniques that are used for the reconstruction of non-periodic objects [3, 4], such as integrated circuits [5], biological proteins [6], bone tissue [7], and more. In single-frame diffractive imaging, an incident beam illuminates an isolated unknown sample. Object features that are comparable in size to the illumination wavelength cause diffraction and the resulting intensity pattern is subsequently measured on a camera. The phase retrieval algorithm then recovers the lost phase information and reconstructs a discrete representation of the object [8, 9, 10]. For extended objects, multi-frame measurements can be made by scanning a localized illumination across a sample, a method known as ptychography [4, 11]. The uniqueness of the reconstruction is guaranteed by illumination overlap between the multiple measurements, improving the reliability of the reconstruction [12, 13].

The trade-off between single-frame and multi-frame diffractive imaging is that more measurements provide more stringent constraints on object reconstruction at the expense of longer time to acquire the data. Efforts have been made to implement ptychography with single-shot measurements, though they come at the cost of high hardware complexity and low information acquisition efficiency [13, 14, 15]. The search of a single-frame imaging method that retains the reliability and flexibility of multi-frame approach continues.

1.1 Randomized probe imaging

Randomized Probe Imaging (RPI) is a single-frame diffractive imaging method that uses randomized light, rather than a finite support constraint, to generate a unique solution to the phase retrieval problem [16]. The combination of randomized illumination and a band-limiting condition on the object provides enough information in the single-frame diffraction intensity to guarantee a unique solution up to a global additive phase factor. RPI is promising, for example, for time-dependent nanoscale X-ray imaging, since it does not introduce any optics behind the sample, or require any alternations to the sample. It has been shown that RPI can produce high-fidelity reconstructions using gradient descent based iterative algorithms [16]. However, conventional iterative algorithms are computationally expensive and typically do not exploit regularizing priors based on the statistical properties of scattering objects. As a result, it can be challenging to process large volumes of data with these algorithms, and they can have limited performance under low-light conditions.

1.2 Deep k-learning

Here, we propose a deep learning framework – deep k-learning – which is specifically designed to address the issues of computational load and low-light performance for far-field RPI reconstructions. Recently, many deep learning based algorithms have been proposed to solve phase retrieval problems, including reconstructions in tomography [17, 18, 19, 20, 21, 22, 23], ptychography [24, 25, 26, 27, 28], and holography [29, 30, 31, 32]. Compared with conventional iterative approaches, deep learning algorithms can produce moderate quality reconstructions with low data redundancy, high computational efficiency, and low latency [17, 24, 29]. Deep learning methods have been particularly successful under noisy, low-light conditions [33, 34].

In most previous works, a deep neural network (DNN), typically a convolutional neural network (CNN), is trained with examples of objects and their corresponding diffraction patterns. The goal is to minimize the loss between the generated objects output by the

network and the ground truth. After training, the network will have learned the direct transformation from measurement to scattering object, implicitly incorporating the physics of light propagation. This is known as End-to-End training, and it relies on the idea that a learnable transfer function exists which maps the intensity measurements onto the object domain. In contrast, deep-k-learning uses an approximated version of the object – the output from one iteration of an iterative algorithm – as the input to the neural network. This follows a recent thread of research that leverages approximate physical operators to generate an input image, also referred to as the "Approximant", which is already in the object domain [33, 35, 36, 37, 38], generally finding significantly improved results even with simpler neural network architectures.

The use of an approximate physical operator has three main advantages over an End-to-End approach. First, the network no longer needs to learn the diffraction physics, which allows for leaner and simpler network architectures. Second, weight-sharing convolutional layers are not well suited to learning maps between the far-field and object domains. This is because the inductive bias in a convolutional layer assumes that relationships between input and output features are local and translationally equivariant. When mapping between far-field and object domains, these assumptions are emphatically not true. Third, pre-trained models and transfer learning can be applied when the network's inputs and outputs follow a natural image distribution, allowing for major speedups when training domain specific models.

Chapter 2

Principle of RPI for 2D imaging

2.1 The experimental geometry of RPI

The basic theory of the RPI was developed by Abe Levitan. The experimental geometry of RPI is outlined in Figure 2-1. A randomized zone plate first focuses coherent illumination at a wavelength λ to a focal spot. An order selecting aperture blocks unwanted higher order diffraction from the zone plate, producing an aperture filled with a band-limited random field at the sample plane. The randomized probe $P(x, y)$ then interacts with a thin sample described by a complex object function $O(x, y)$. In our work, we consider phase-only objects $O(x, y) = \exp(i\phi(x, y))$ for simplicity. The resulting exit wave $E(x, y) = P(x, y)O(x, y)$ propagates to the Fraunhofer plane where its intensity is measured by a charge-coupled device (CCD) camera. The noiseless intensity measurement $I_0(k_x, k_y)$ thus can be written as

$$I_0(k_x, k_y) = |\mathscr{F}\{P(x, y)O(x, y)\}|^2, \tag{2.1}$$

where $\mathscr{F}$ denotes the Fourier transform operator when the exiting wave propagates to the far-field. In practice, measurements are also subject to various sources of corrupting noise such as Poisson statistics with parameter λ due to the discrete nature of light and $\mathcal{N}$ is the additive Gaussian noise due to thermal fluctuations at the photoelectric detector circuitry..

We express the noisy measurement $I(k_x, k_y)$ in the far-field as

$$I(k_x, k_y) = \mathscr{P}\{I_0(k_x, k_y)\} + \mathscr{N}, \tag{2.2}$$

where $\mathscr{P}$ denotes Poisson sampling with parameter λ and $\mathscr{N}$ is the additive Gaussian noise.

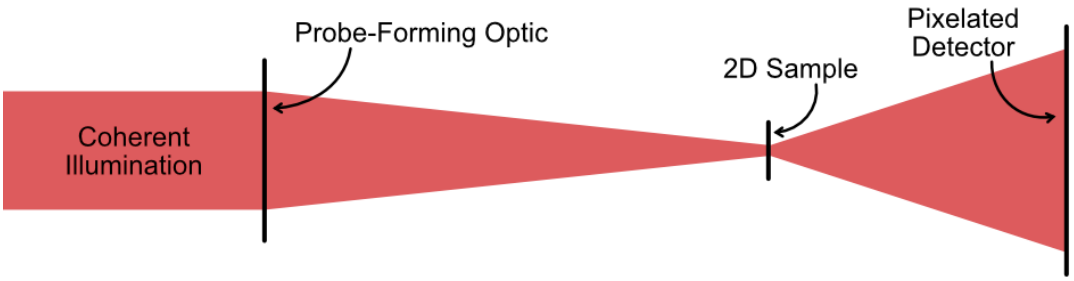


Figure 2-1: A conceptual diagram of the layout used in an RPI experiment.

In the RPI reconstruction process, the measured single-frame diffraction intensity $I(k_x, k_y)$ and prior knowledge of the probe wavefield $P(x, y)$ are used to reconstruct a discrete representation of the object $O(x, y)$. Note that the presence of randomized illumination $P(x, y)$ breaks the spatial shift and conjugate inversion degeneracy of the classic two dimensional Coherent Diffractive Imaging (CDI) problem [39]. Rather than resorting to a finite support constraint as in traditional CDI, the reconstruction process in RPI uses a band-limiting constraint on the object to restrict the number of free parameters and achieve sufficient data redundancy.

Importantly, this reconstruction process is only well-posed when the diffraction pattern contains a sufficient amount of measurements in excess of the number of independent degrees of freedom in the object. Without additional information about the object, this leads to an expectation that a stable reconstructions can be achieved when the highest frequency k_p at which the probe has nonzero power remains larger than the frequency k_o to which the object is band-limited. Based on this analysis, it is useful to define the resolution

ratio $R = \frac{k_o}{k_p}$ [16]. As the resolution ratio decreases, the sampling redundancy increases, producing more stable (but lower-resolution) reconstruction. Here, we consider the role of machine learning approaches at various values of R, ranging from low values (~ 0.25) where the reconstruction is extremely tightly constrained to high values (~ 2) where, without additional information, the problem is almost certainly ill-posed.

2.2 End-to-end phase retrieval

Convolutional neural networks are an indispensable tool for many modern computer vision applications, such as image classification [40], objection detection [41], and neural style transfer [42]. Many recent works have also shown that convolutional networks perform well in solving phase retrieval problems [33, 43, 34, 44, 45, 35].

The most basic way to apply a convolutional neural network to the phase retrieval problem, which remains the basic standard, is known as the End-to-End approach. In this design, one trains a network using the raw diffraction patterns as an input, producing as output an estimate of the retrieved object. In our case, this output would be an estimate of the phase of a thin, phase-only object. This works well, or at least acceptably, for many variants of diffractive imaging based on Fresnel propagation [33, 35, 34, 46].

Considering the design of a standard convolutional network, outlined in Fig. 2-2, can help us understand why these networks are a natural fit to Fresnel-based phase retrieval problems.

U-style type of networks are divided into an encoding arm and a decoding arm. The encoding arm learns to predict a representation of the scattering object in a low-dimensional latent space based on an input diffraction pattern. The decoding arm learns to map from the embedding manifold back to the discrete representation of the scattering object - the desired final result. Often, skip connections are used to bypass the feature maps from the encoder arm to the the corresponding layers in the decoders arm. This allows local information to be transferred directly from the input to output domains, which helps preserve high frequency structures in the reconstruction [46].

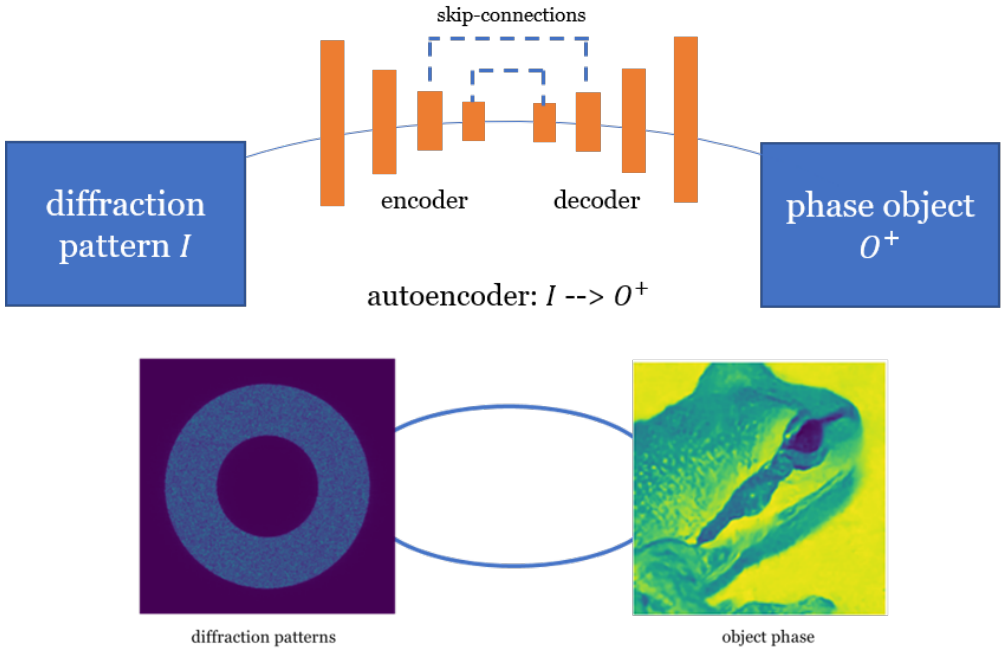


Figure 2-2: Architecture of conventional encoder-decoder

Convolutional neural networks often work well when the relationship between the input and output domains is fundamentally local. This is because weight-sharing convolutional layers preserve translation equivariance [47], such that a shifted input to a layer produces a shifted output. Because Fresnel diffraction patterns do preserve the location of features in the scattering object, the map that must be learned to perform phase retrieval naturally shares the same translation equivariance as the convolutional layers.

However, convolutional networks are not ideal when the input diffraction patterns are in the far-field regime (as is the case for RPI), for two major reasons. First, the real-space to Fourier space mapping is global. In a far-field phase retrieval such as RPI, every pixel in the diffraction pattern includes a contribution from every pixel in the real-space object domain. Second, the real-space to Fourier space mapping does not respect translation equivariance. A shifted input diffraction pattern should be mapped to a version of the output object with linear phase ramp, rather than a translated version of the corresponding output object. This is formalized with the following inequality:

$$g_0(x+\delta x, y+\delta y) \neq |\mathscr{F}\{P(x,y)O(x+\delta x, y+\delta u)\}|^2. \tag{2.3}$$

Chapter 3

Our solution: deep k-learning

3.1 Physical operator and autoencoder with 2D Conv

The workaround we used for applying convolutional neural networks to phase retrieval in the far-field (in this case, RPI) is to apply an approximate map from the diffraction pattern domain to the object domain *before* using the neural network for the final reconstruction. This framework is depicted in Fig.3-1. Although the approximate map cannot produce an accurate reconstruction on its own, it creates inputs for training and inference which already live in the same image space as the final reconstructed objects. We call this approach deep-k-learning, because it is designed to compensate for the issues created by having input data which are organized in k-space.

The choice of approximate mapping is clearly of crucial importance. Here, we use a single iteration of a gradient-descent based iterative algorithm solving the following optimization problem for a diffraction pattern I_i:

$$\mathbf{O}_i^+ = \underset{\mathbf{O}_i'}{\operatorname{argmin}} \mathscr{L}\left\{I_i, |\mathscr{F}\{P \times O_i'\}|^2\right\} \tag{3.1}$$

Here, O_i' is a low-fidelity estimate of the band-limited object and P is the known probe state. The probe P is either known *a priori* (as for simulation), or retrieved via ptychography measurement (as in the experiment). The output of a single step of the iterative algorithm

when initialized with a uniform object is called the Approximant and is denoted by O_i^*. When more steps of the optimization is taken (with lower learning rate), we regard the output as iterative reconstruction. Adam optimizer is chosen to generate Approximant and iterative reconstruction for fair comparisons. The Approximant is then fed into a CNN based autoencoder $G_{\mathbf{w}}$ with parameters $\mathbf{w}$. The training process learns a map from Approximants to ground-truth objects, formally written as:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \mathscr{L}\{O_i, G_{\mathbf{w}}(O_i^*)\} \tag{3.2}$$

where the optimal weights after gradient descent are $\hat{\mathbf{w}}$, and O_i is the ground truth object.

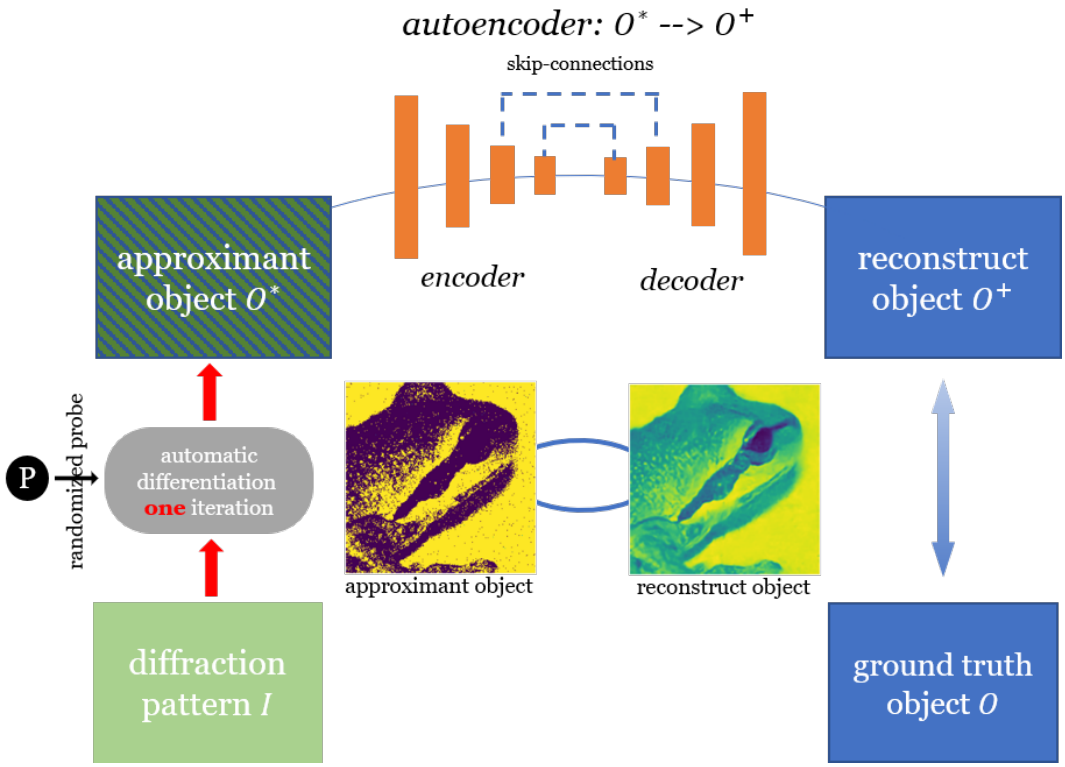


Figure 3-1: Our deep k-learning framework

The network we used is an autoencoder architecture [48], where the encoder arm whose architecture is similar to that of EfficientNetB7 [49] to enable efficient feature extraction. The feature pooling is built on inverted residual blocks (or MBConv), where the input and output

of the residual block are thin bottleneck layers as opposed to traditional residual blocks to achieve efficient feature extraction [49, 50]. In each inverted residual block, convolutional layers are being deployed to extract local features, and squeeze and excitation (SE) blocks are being used to extract global features [51]. Note that the convolutional layers in our implementation are combined with depth-wise and point-wise convolutions to reduce the computation cost [52]. Residual connections within each block are employed to avoid the problem of vanishing gradients [53], batch normalization is adopted to stabilize the learning process [54], and dropout layers are used to prevent over-fitting. Down-sampling in the encoder arm is achieved via average pooling block by block, with a pooling size of (32, 32) in total. Therefore, the final embedded output from the encoder has a dimension of (H/32, W/32, C), where H and W are the height and width of the input object, and C is the channel size in the last inverted residual block. In our implementation, C is 2560.

The decoder arm is comprised of five residual up-sampling blocks with up-sampling. The up-sampling is achieved by transposed convolution. Each up-sampling transposed convolution layer is followed by two convolution layers with same filter and kernel sizes. The scaling factor of all the up-sampling blocks is (32, 32) in total, producing an output with the shape (H, W, 1). Skip connections are used between encoder and decoder arms to preserve high-frequency information [46]. The detailed network architecture can be found in our github page.

3.2 Supervised representation and adversarial loss

Three main choices exist for the loss function $\mathscr{L}$ needed to train the deep k-learning framework: supervised loss, representation loss, and adversarial loss. Here, we constructed a loss function consisting of a mix of all three types. When the network is trained with a mix of all three types, we call it generative deep k-learning. When the network is trained with supervised loss only, we call it non-generative. The supervised loss, which directly compares predicted ground truth objects, is the main component. In our implementation, supervised loss was implemented as the negative Pearson correlation coefficient (NPCC) between the

reconstructed objects and the ground truth, defined as

$$\text{NPCC} = -r_{X,Y} = -\frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}, \tag{3.3}$$

where cov is the covariance and σ_X, σ_Y are the standard deviations of X and Y, respectively. Previous works have shown that NPCC is more effective in recovering fine features than pixel-wise loss functions [34, 46, 55, 35]. In the context of our network, NPCC is written as:

$$\mathscr{L}_{\text{npcc}}(G_{\mathbf{w}}) = \mathbb{E}_{O,O^*}[-r_{O,G_{\mathbf{w}}(O^*)}] \tag{3.4}$$

To define the representation loss, we use an ImageNet pretrained EfficientNetB0. This representation loss is a perceptually-motivated loss which measures the mean absolute error between the latent space representation of the reconstructed object $H(O^+)$ and the embedding of the ground-truth object $H(O)$. Here, H refers to the pretrained EfficientNetB0 encoder. It may improve the reconstruction quality without changing the network architecture [56], helping the generative model to synthesize features closer to the ground truth distribution. In our implementation, we choose L1 or mean absolute error to measure the distance between the two distributions:

$$\mathscr{L}_{\text{mae}}(G_{\mathbf{w}}) = \mathbb{E}_{O,O^*}[\|H(O) - H(G_{\mathbf{w}}(O^*))\|_1] \tag{3.5}$$

The adversarial loss is computed with a CNN-based discriminator. Our implementation of adversarial loss is inspired by conditional generative adversarial networks (cGANs), a particular training strategy that uses a discriminator to compete with the autoencoder/generator [57, 58, 59]. The objective of cGAN for our RPI problem can be written as follows:

$$\mathscr{L}_{\text{adv}}(G_{\mathbf{w}}, D'_{\mathbf{w}}) = \Big(\mathbb{E}_{\mathbf{o}\sim\mathbf{p_o}(\mathbf{o})}\big[\log D_{\mathbf{w}'}(\mathbf{o})\big] + \mathbb{E}_{\mathbf{o}^*\sim\mathbf{p}_{\mathbf{o}^*}(\mathbf{o}^*)}\big[\log(1 - D_{\mathbf{w}'}(G_{\mathbf{w}}(\mathbf{o}^*)))\big]\Big) \tag{3.6}$$

Now, our autoencoder becomes a generative model G that tries to generate objects with the

highest possible value of $D(G(\mathbf{o}^*))$ to fool the discriminator D, as shown in the second term of Eq. 7.4. Simultaneously, the discriminator D tries to maximize its ability to recognize ground truth objects as real and generated objects as fake. This component of the loss updates the weights in the discriminator. During training, the generator and discriminator are simultaneously updated based on their respective losses. The adversarial loss generally is thought to help the autoencoder/generator learn the transformation of the noise within the object Approximant to plausible features in the final reconstructed object, given the prior of ground truth distribution O.

Finally, the total loss for the generator of our deep k-learning framework is defined as:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{npcc}}(G_{\mathbf{w}}) + \alpha \times \mathscr{L}_{\text{mae}}(G_{\mathbf{w}}) + \beta \times \arg\min_{G_{\mathbf{w}}} \max_{D_{\mathbf{w}'}} \mathscr{L}_{\text{adv}}(G_{\mathbf{w}}, D'_{\mathbf{w}}) \tag{3.7}$$

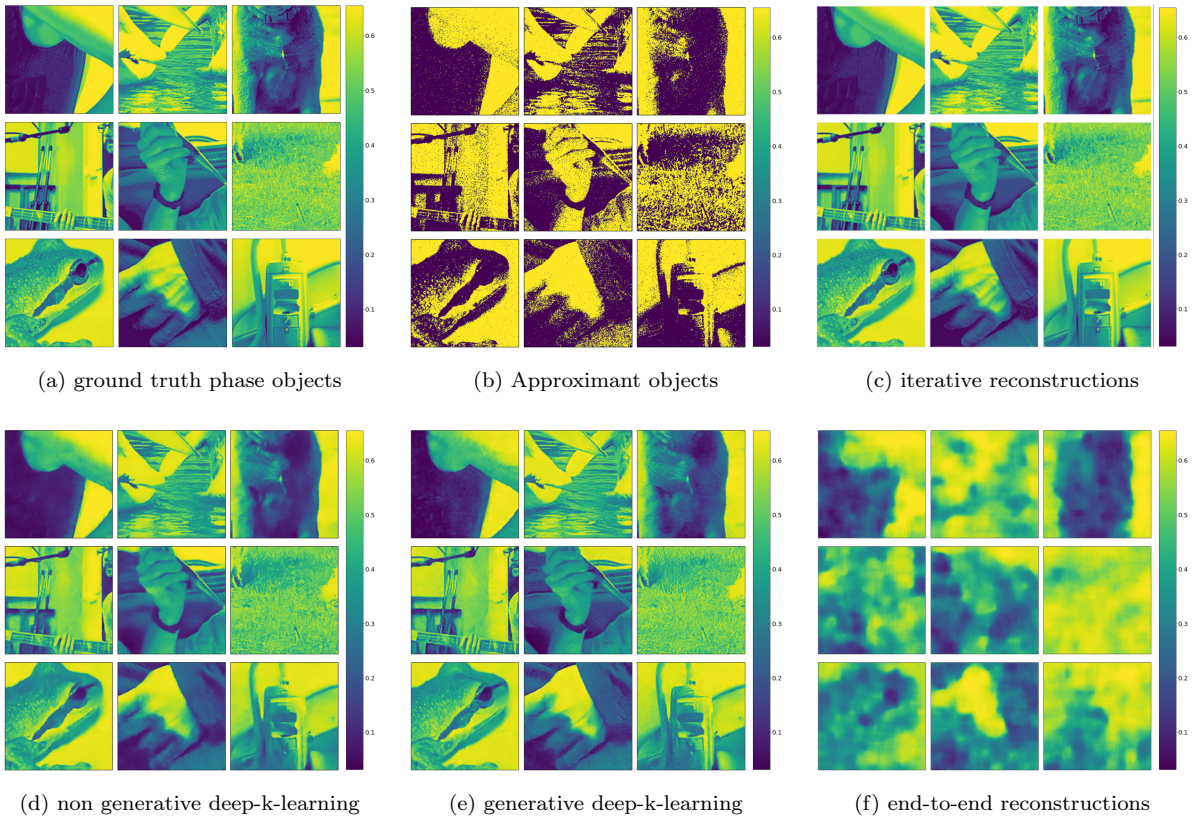
Here, α and β are hyper-parameters that determine the relative weights between the three types of learning loss. For the non-generative framework, α and β are set to zero.

Chapter 4

Numerical results for 2D imaging

We conducted a set of numerical simulations to demonstrate the effectiveness of the deep k-learning method on the RPI phase retrieval problem. We focused on the role of the resolution ratio R and the noise level. High resolution ratios R and low signal regimes are particularly interesting to study because these conditions are the most challenging scenarios for iterative algorithms, and therefore are most likely to benefit from the added information about the object distribution that deep-k-learning can introduce.

In our first experiment, we studied the performance of the various proposed methods under ideal illumination conditions. We simulated an RPI experiment using 256×256 pixel objects defined with uniform amplitudes and phases drawn from randomly cropped ImageNet images, scaled to a range of up to 1 radian. 4,000 training examples and 100 testing examples were simulated, at $R = 0.5$ with 10^4 photons per pixel in the 256×256 pixel object. Fig. 4-1 shows a visual comparison between the phase images reconstructed with each method. Fig. 4-1(a) shows a set of ground truth objects selected from the testing dataset. In Fig. 4-1(b), the corresponding Approximants are shown. We can see that the approximate map successfully retrieves the general structures of the object, albeit at an incorrect overall scale. Additionally, noise and artifacts are readily apparent and, when just considered as images, the Approximants are of low quality. In contrast, Fig. 4-1(c) shows the converged results from iterative reconstructions after 100 iterations. Visually, they look identical to the ground



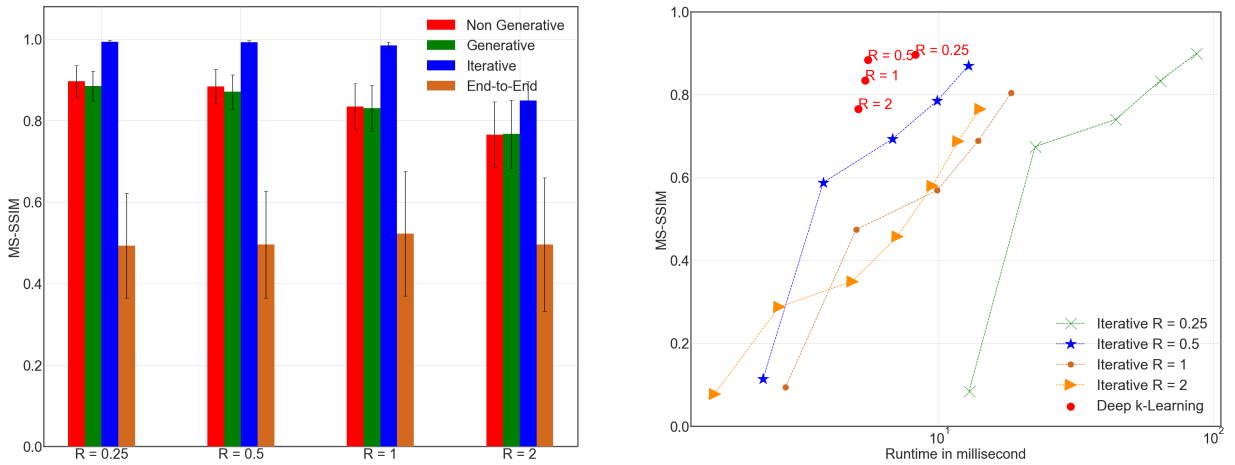
(a) ground truth phase objects (b) Approximant objects (c) iterative reconstructions

(d) non generative deep-k-learning (e) generative deep-k-learning (f) end-to-end reconstructions

Figure 4-1: Visual comparison for the phase-only object reconstruction at $R = 0.5$ with 10^4 photons per pixel. The color bar is set to the range of the ground truth images. (a) contains the ground truth phase-only objects, (b) contains the input Approximant with one iteration, (c) contains the iterative reconstructions, (d) contains the non generative deep-k-learning reconstructions, (e) contains the generative reconstructions, (f) contains the end-to-end reconstructions.

truth phase objects, as expected based on the ideal imaging conditions [16].

Moving to the neural network outputs, Fig. 4-1(d) shows the non generative deep k-learning reconstructions. Drastic improvements are obvious when compared with the input Approximants. Reconstruction are now smoother and contain fine details that were washed out by noise in the input Approximants. However, although the results have high visual quality, there are noticeably missing fine features when compared with ground truth and iterative reconstructions. Fig. 4-1(e) has the generative deep-k-learning reconstructions, although visually the difference between non-generative and generative reconstructions under these illumination conditions is not obvious. Finally, Fig. 4-1(f) contains the output of the end-to-end network reconstructions. These results only contain low frequency information about the phase objects. This is not entirely unexpected, given the previous arguments about the mismatch between convolutional neural networks and mappings between k-space and real-space.



(a) simulation results for R from 0.25, 0.5, 1, to 2

(b) runtime comparison at different R values

Figure 4-2: Quantitative comparison between different training frameworks at different R

4.1 Performance dependency on oversampling ratio

After confirming that deep-k-learning is capable of producing moderate quality images under ideal conditions, we studied how its performance depends on the relationship between the

highest frequencies in the object and those in the probe. In Fig. 4-2(a) we show a quantitative comparison of reconstruction quality at values of R ranging from 0.25 to 2 at 10^4. The x-axis represents the resolution ratio R, and the y-axis reports the MS-SSIM (Multi-scale Structural Similarity) metric for the reconstruction quality. The reported value is the mean MS-SSIM result over the test reconstruction set, and the error bars show the standard deviation within the test dataset. Recall that larger values of R describe more challenging conditions where the features in the object are smaller when compared to the speckle size in the probe.

A total of four reconstruction methods are reported in the figure: non generative deep k-learning, generative deep k-learning, an iterative algorithm (100 iterations), and the End-to-End training method. Details about the data processing and network training can be found in the Appendix. Iterative reconstructions have the best performance over the full range of R. At $R \leq 1$, the MS-SSIM evaluations of iterative reconstructions approach 1, as expected [16]. As R increases beyond 1, the iterative reconstructions start to degrade. These observations agree with previous work, as larger values of R lower the data redundancy in the diffraction patterns. Both variants of deep k-learning methods outperformed end-to-end networks, although the results still underperformed the iterative reconstructions. Reconstruction quality also degraded with R across methods, as expected. However, the End-to-End reconstructions' quality plateaus at a lower value of R, regardless of the oversampling ratio, in agreement with the visual observations.

Although the deep-k-learning reconstructions do not produce the same level of fidelity as the iterative results, they run much faster. Fig. 4-2(b) compares the per-pattern runtime of iterative algorithms and deep k-learning method across R. The intermediate results from the iterative reconstructions are shown at 1, 5, and 10 iterations, and every 10 iterations thereafter until they surpass the comparable deep-k-learning result. The strong dependence of per-iteration runtime on R arises because smaller values of R require more highly textured probes, which are stored in larger arrays. These results reveal that the deep k-learning results have comparable quality with iterative reconstructions at around 40 to 50 iterations. However, the computational speedup provided by deep k-learning ranges from 3x to 10x,

depending on the value of R.

4.2 Performance dependency on low-light noise

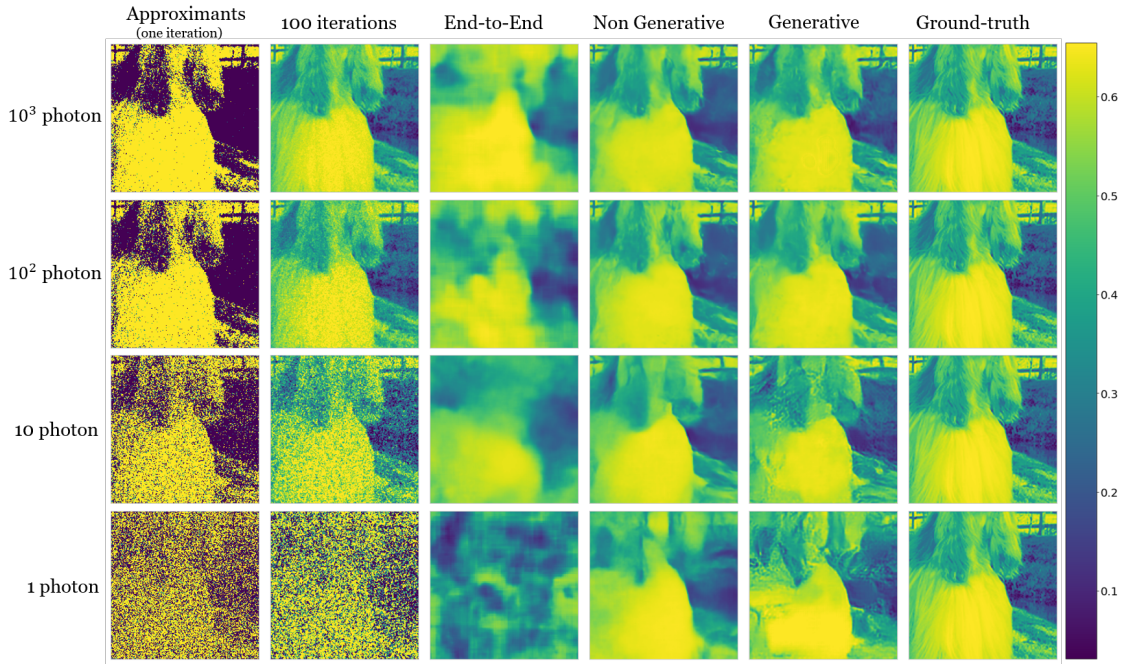
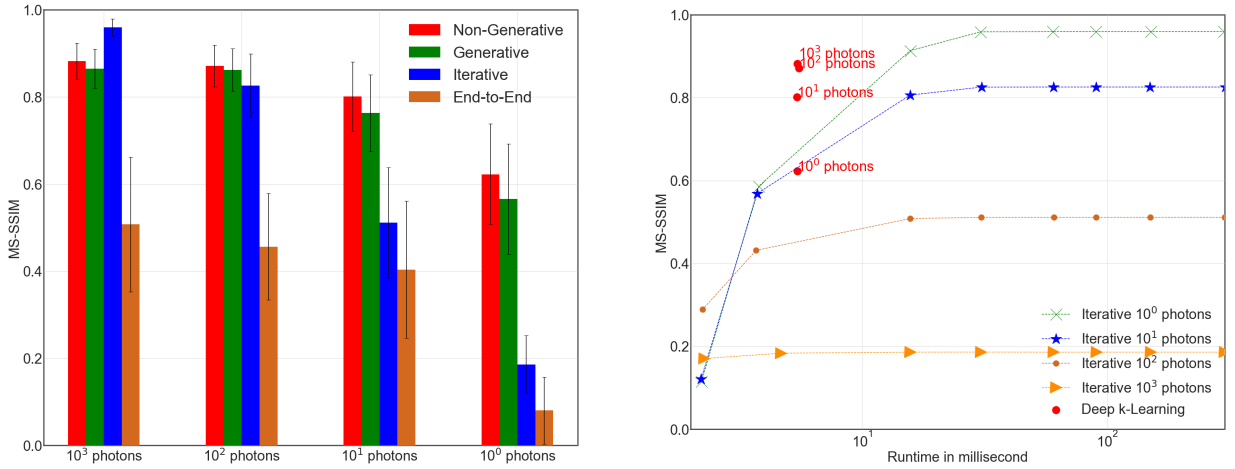


Figure 4-3: Visual comparison for the phase-only object reconstruction for R=0.5 at low photon imaging conditions. The colorbar is set to the range of the ground truth images.

Finally, we investigated the performance of deep-k-learning for RPI under noisy conditions, where knowledge of the object's prior statistics is the most valuable. Fig. 4-3 shows a visual comparison for the phase images reconstructed at $R = 0.5$, with illumination levels ranging from 10^3 to 1 photon per object pixel. As the photon incidence rate decreases, reconstruction quality inevitably decreases as well. As expected, the iterative reconstruction quality is strongly dependent on the photon shot noise level, with the signal quickly fading under growing background. However, the deep-k-learning results generally retain their visual quality even at photon rates low enough to cause the iterative method significant degradation. In the single photon case, the iterative reconstruction becomes nearly unrecognizable, while both deep k-learning methods produce reconstructions that, while visibly degraded,

maintain many of the general features of the object.



(a) results for R = 0.5 at low photon conditions

(b) runtime comparison at low photon conditions

Figure 4-4: Quantitative comparison between different training frameworks at low photon imaging conditions

Fig. 4-4(a) shows the quantitative comparison from the same sweep over low photon imaging conditions, following the same format as Figure 4-2. These quantitative results confirm the analysis from our visual inspection of Fig. 4-3. Both deep k-learning methods are significantly more robust to Poisson noise than the iterative methods, producing reconstructions with superior quality starting at 10^2 photons. As the photon number decreases further, the gap between deep k-learning and iterative reconstruction quality grows. This shows the effectiveness of the strong object prior embedded in the deep k-learning methods through the training process.

Finally, in Fig. 4-4(b) we consider the runtime speedup available under high noise conditions, comparing the iterative algorithm with the best variant of deep-k-learning method at each imaging condition. Due to the feed-forward nature of deep learning, deep k-learning takes under 10 milliseconds to produce each result, while the iterative algorithm require around 100 milliseconds to converge, suggesting that the 10x speedup under ideal illumination is preserved, or even improved upon, under adverse, noisy conditions.

Overall, our simulation results show that deep k-learning is both faster and more robust to Poisson corruption than the iterative algorithm. Particularly when photon levels reach

10^2 photons per object pixel or lower, deep k-learning outperforms iterative algorithms in terms of reconstruction quality with much faster computational speed.

Chapter 5

Experimental results for 2D imaging

5.1 Experimental setting

To demonstrate that the deep k-learning approach can successfully be translated from simulation to experiment, we performed phase retrieval with deep k-learning on a large dataset of RPI diffraction patterns collected from an optical table-top apparatus. The experimental apparatus was constructed in collaboration with Abe Levitan. To draw test images from a well understood distribution, we used a Spatial Light Modulator (SLM) to display 256x256 phase-only images randomly drawn and cropped from the ImageNet dataset. The experimental design is diagrammed in Figure 5-1.

Polarized light was generated by passing a 635 nm laser diode source (Thorlabs CPS635F) through a film polarizer aligned to the optic axis of the SLM. This light was then spatially filtered by a 5 μm pinhole at the focus of a beam expander to enforce spatial coherence across the beam diameter. A randomized pattern was then imprinted on the wavefield using a randomized zone plate with a 2 cm diameter and a 50 cm focal length, producing a focal spot with an overall diameter of 2 mm. An adjustable iris acted as an order selecting aperture for this diffractive optic.

The focus of the randomized zone plate was aligned to the plane of a reflective SLM (Thorlabs EXULUS HD2) at normal incidence. The phase-only SLM consisted of pixels

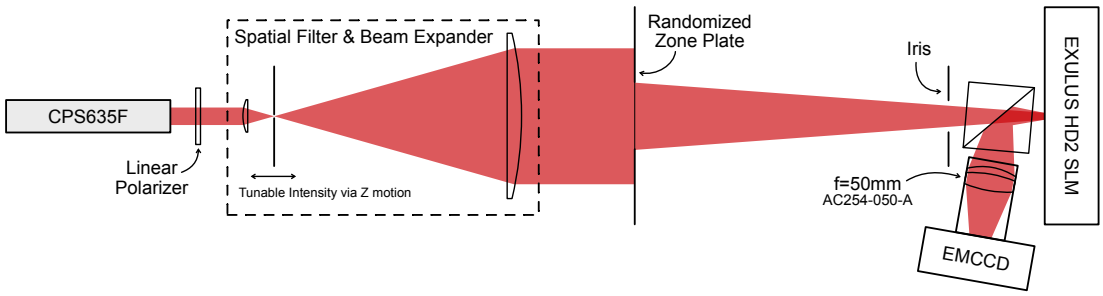


Figure 5-1: A diagram of the experimental design for our tabletop demonstration.

arranged with an 8 μm pitch, each of which imprints a variable phase delay between 0 and 2π on the light field. The reflection was then separated with a non-polarizing 50/50 beamsplitter cube placed approximately 5 degrees from normal to prevent higher order reflections from overlapping with the primary beam on the detector. The Fourier plane was finally imaged on a EM-CCD camera (QImaging Rolera EM-C2) with 8 μm pixels, placed at the focus of an achromatic doublet with a 50 mm focal length (Thorlabs AC254-050-A). A 992×992 pixel region was cropped from the detector, such that the real-space grid corresponding to the measured slice of reciprocal space consists of 8 μm pixels, aligned with the pitch of the SLM.

5.2 Experimental reconstructions

We collected four datasets under different imaging conditions, targeting photon fluxes of 1, 10, 100, and 1000 photons per pixel in the 256x256 object. The CCD was calibrated to allow conversion between analog-digital units (ADUs) and photon counts. A detailed summary of the experimental measurements, including more information on the noise properties of the detector, can be found in the Appendix. For each imaging condition, we initially collected a ptychography dataset on a standard test image (cameraman) to calibrate our knowledge of the probe state. Once calibrated, we collected a set of $4,000$ training and 100 test diffraction patterns from the cropped ImageNet objects. In each case, the images were first converted to 8-bit greyscale images, and finally displayed on the SLM such that the full 8-bit range

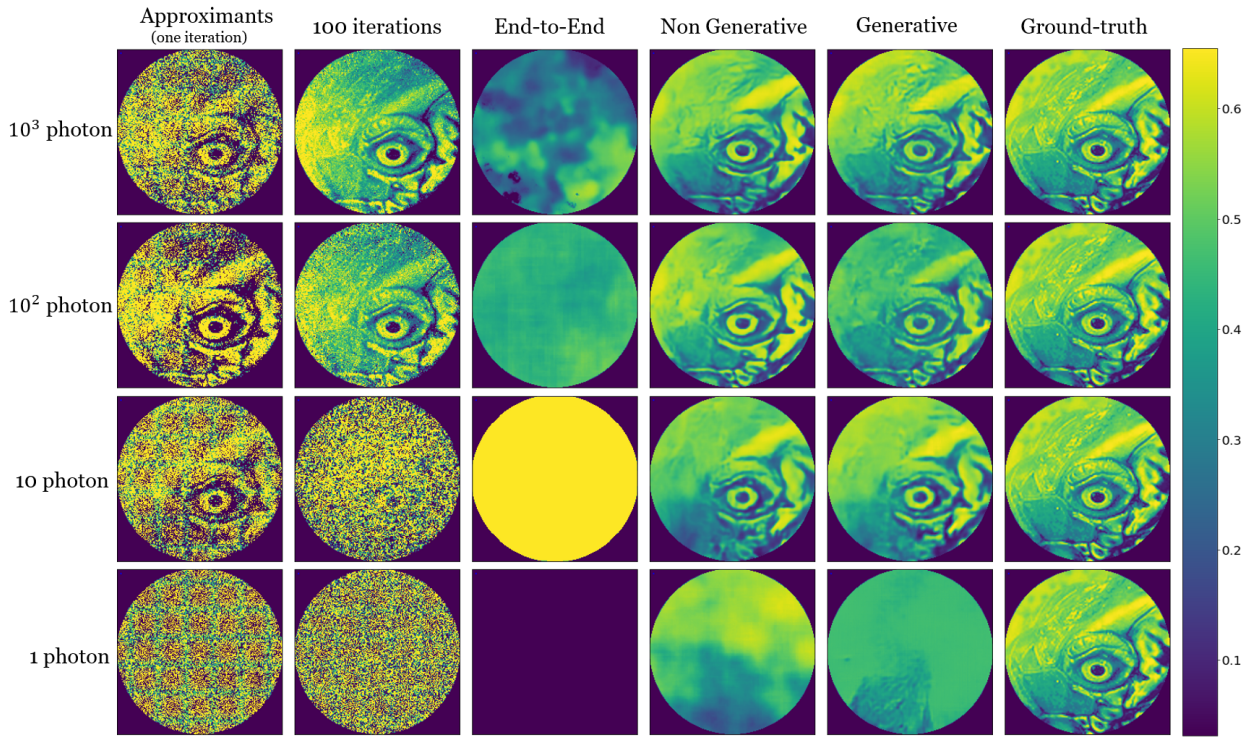


Figure 5-2: Experimental reconstruction comparison between different methods under low photon conditions. The colorbar is set to the range of the ground truth images.

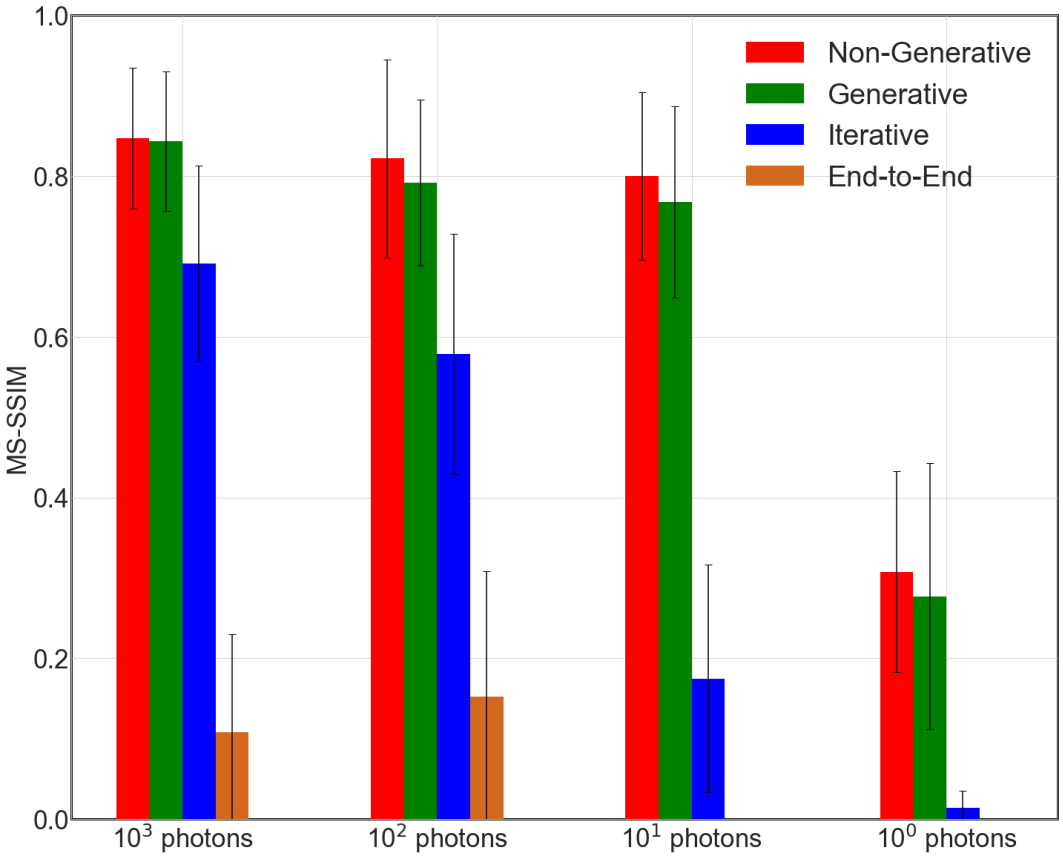


Figure 5-3: MS-SSIM comparison between deep-k-learning and iterative algorithm on different Poisson noise corrupted imaging conditions

corresponded to a sweep from 0 to 2π radians. Details about experimental measurements can be found in the Appendix.

In Fig. 5-2 we show the visual comparisons of the test set between different reconstruction algorithms under low photon conditions. These reconstructions are produced with the same set of algorithms we studied numerically. The most visible difference between simulation and experiment is that, while in our simulations we assumed the randomized probe focal spot covers the entire field-of-view, in the experiments the objects are illuminated by a finite circular probe. Thus, the edges of the object window are not illuminated by the probe and thus do not contain any object features.

Near 10^3 photons/pixel, iterative reconstructions show good results in the central region, getting more and more noisy toward the weaker edge of the probe. This is due to the spatial variation of the illumination intensity profile. Both the non generative and generative deep k-learning produce visually high quality reconstructions over the entire field of view of the probe, although the networks minor artifacts are indeed introduced, especially at the lower end of the photon incidence rates.

As we decrease the photon budget down through 100 to roughly 10 photons per object pixel, the quality of the generative reconstructions slowly decreases while the noise rapidly takes over and dominates the iterative results. The End-to-End model begins to diverge at 10 photons per object pixel. As we lower the signal rate further, to 1 photon per object pixel, the reconstructions from all methods fail. To account for the disparity, it is important to recognize that due to the presence of readout noise and other non-Poisson sources of noise, the signal to noise ratio of these images is far lower than that of our simulated dataset at 1 photon per pixel.

Fig. 5-3 shows a numerical comparison that confirms our observations. Note that we only include the illuminated region (the region in the center images of Fig. 5-2) when computing the MS-SSIM values for each method. For iterative reconstruction, we also shift the output pixel to compensate for a slight misalignment in our optical system. Compared with simulation results, deep k-learning methods maintain a moderate quality level in the range

of 0.8 under the second-to-lowest illumination conditions, while the quality of the iterative results deteriorates as the photon number decreases. End-to-end MS-SSIM drops to near zero starting at 10 photon per object pixel, as the pixel values of the outputs are outside the range of the ground truth. These results suggest that the iterative algorithm is much more prone to noise degradation than the deep-k-learning approach. Thus, deep-k-learning emerges as a valuable alternative particularly under noisy conditions. This is because under such noisy experimental conditions the deep k-learning algorithm is far more effective at incorporating strong object priors to regularize the reconstructions.

Chapter 6

Multi-frame 3D imaging via tomography

6.1 X-ray tomography

X-ray tomography is a powerful method for imaging the internal details of objects in three dimensions non-invasively [60, 61, 62], and it has wide applications in biomedical imaging [63], materials study [64], electronic inspection [65], and other fields. The penetrating ability of X-rays makes it possible to obtain a series of two-dimensional Radon transforms (commonly known as radiographs) of the object viewed from different angles [66]. After capturing radiograph measurements, objects can be reconstructed using a three-dimensional computed tomography algorithm.

The reconstruction process of X-ray tomography is generally an ill-conditioned inverse problem. This is because measurements taken at a discrete number of angles can only sparsely sample the high frequencies of the object. Therefore, full-angle measurement with high sampling rate is preferred to best resolve the ambiguity in the inverse solution. However, in practice, limited-angle measurement is often used due to the long time of acquiring the full angle measurement, leaving entire sectors of the Fourier space unsampled [67, 18, 68, 69, 70, 71]. For objects or samples that are radiation-sensitive, a low photon-budget per scan is also preferred to minimize the total exposure and potential damage, making the effect from ill-conditioning even more severe.In such cases, an analytic reconstruction algorithm

like filtered back projection (FBP) is inadequate as it can generate reconstructions with noise and streak artifacts [72]. Iterative algorithms whose objective function includes a term representing prior knowledge about the object may compensate for the deficits in Fourier space coverage and thus often produce higher fidelity results [73]. This is understood as regularizing the problem, *i.e.,* reducing the space of possible solutions of the inverse problem to a subdomain in which the object must belong [74, 75]. When prior knowledge is used in an iterative algorithm, the optimization balances minimization of the residual of the simulated measurements from a reconstructed object against minimization of the regularization term. Assumed priors such as sparsity, total variation, and nonlocal similarity priors have shown promising results for X-ray tomography [76, 77, 78]. However, without trial and error, it is not straightforward to choose the appropriate prior and regularization weight for a given set of objects [68]. A prior distribution may also be learned from the dataset itself by a machine learning algorithm. Using a large amount of paired training data, a prior can be determined through exploring the statistical properties of the training distributions, improving the reconstruction quality. Recently, learned priors have been successfully applied to tomography in treating the ill-conditioned inverse problem. In particular, deep learning, a subset of machine learning that is based on artificial neural networks, achieved promising results [67, 18, 68, 69, 70, 71]. For example, efforts have been made in using learned priors from deep neural networks to recover boundary information [70], and to generate missing projections with a data-consistent reconstruction method [71]. However, some works have shown that these methods suffer from reconstruction vulnerabilities and instabilities [79, 80]. To avoid these issues, a recent thread of research leverages reconstructions from an analytic or iterative algorithm [68, 69, 81, 23, 22], or uses a two-step deep learning strategy to generate reconstructions that are empirically more stable and accurate [82].

6.2 Physics-assisted Generative Adversarial Network

Here, we develop a Physics-assisted Generative Adversarial Network (PGAN) for limited-angle X-ray tomography, and demonstrate the ability of the learned prior in imaging the

structure of 3D integrated circuits at low photon conditions. In contrast to the previous efforts, our PGAN utilizes a maximum-likelihood estimate (MLE) with a physical prior to compensate for the inherent ill-conditioning of the problem, especially the prevalence of shot noise of Poisson statistics in the low-photon measurements. This physics-informed MLE is then input to a trained deep generative model to produce improved reconstructions. Therefore, the PGAN reconstruction is generated by leveraging knowledge of the Poisson shot noise process in the iterative algorithm and learned prior from deep learning. To evaluate our reconstruction method, we propose a model dubbed CircuitFaker to produce synthetic circuits that are capable of emulating real-world integrated circuits (IC) with design rules. The implicit correlations of the circuits constitute the prior to be assumed or to be learned for the reconstruction algorithms. We simulate X-ray imaging using projection approximation with consideration of attenuation only. Then, we formulate four different variants of deep generative models, using the maximum-likelihood estimate from an iterative algorithm as the input approximant to include the physical priors. The output of the generative models is the reconstructions that have been regularized by the learned prior. We show that the learned prior from the deep generative models dramatically improves the reconstruction quality compared to maximum-likelihood estimation if the photon flux is limited. The key result here is demonstrating that deep learning enables reductions in the total photon budget while retaining reconstruction fidelity.

6.3 Forward model for X-ray tomography

X-ray tomography is an imaging technique to resolve a three-dimensional object non-invasively. Its imaging system usually consists of an object holder, an objective zone plate, and a detector, and the illumination is generated from an X-ray source. The measurements are taken from a series of rotation angles of the object of interest, where a cone-beam geometry is generally assumed to produce the ray projection from the source point to the object, and then from the object to the center of each detector pixel. A conceptual diagram for the imaging system is in Fig. 6-1, here the object is a three-dimensional IC. In the absence of

noise, the exact detection model is written as

$$g^{(0)} = \int dE\, D(E)\, I^{(0)}(E) e^{-\alpha(E) A f} \tag{6.1}$$

Here, A is the system matrix, *i.e.,* the distance each ray traverses from the source through the object to a detector pixel, where each column corresponds to one sample voxel and each row corresponds to one detector pixel over all sample rotation angles, f is the vector of voxel compositions of the object (dielectric or copper for IC), E is the photon energy, $\alpha(E)$ is the absorption coefficient at energy E for copper, $I^{(0)}(E)$ is the source intensity, $D(E)$ is the detector efficiency, $g^{(0)}$ is the vector of expected number of photons measured for each of the detector pixels. Note that the exponential is applied component-wise. When the illumination is monochromatic, Eq. 6.1 can be simplified to:

$$g^{(0)} = N_0 e^{-\alpha A f} \tag{6.2}$$

where N_0 is a vector containing the expected number of photons in each ray, *i.e.,* the index of the element corresponds to a ray. A is the system matrix, and α is the absorption coefficient as before. This can be written in the form of a linear equation:

$$\ln g^{(0)} - \ln N_0 = -\alpha A f. \tag{6.3}$$

Here, the natural log is defined component-wise. In our numerical simulations, we assume the measured photon counts are Poisson-distributed in low photon imaging.

6.4 Iterative algorithms with prior regularizer

An inverse problem is a task of resolving parameters that cannot be directly observed using a set of measurements [83]. In the context of X-ray tomography, the task is to find a

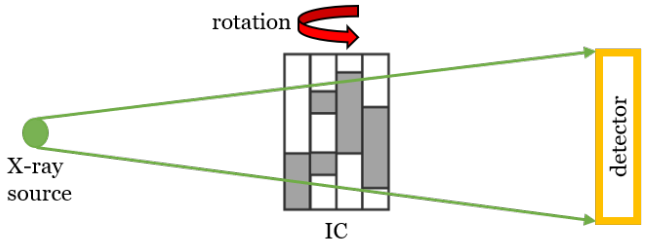


Figure 6-1: A conceptual diagram for our imaging system (IC as the object).

discrete representation of an object's composition f based on the measurements g taken on a digital camera at multiple angles. An analytic reconstruction algorithm such as FBP can solve the inverse problem when the measurements are taken at full-angle and ample illumination. However, when taking measurements with limited angles, with few photons, or a combination of both, the FBP method is not preferred due to the noise and streak artifacts in its reconstruction [72]. The general Wiener-Tikhonov approach [84, 85] improves the analytic reconstruction by solving the following optimization (assuming Gaussian noise in the measurement) iteratively:

$$\hat{f} = \arg\min_{f} \left\{ \| - \alpha A f - \ln g + \ln N_0 \|^2 + \beta \, \Psi(f) \right\} \tag{6.4}$$

where $\hat{f}$ is the inverse result, $\| \cdot \|^2$ is the square of the ℓ^2 norm, $\Psi(f)$ is the regularizer or Bayesian prior, and β is the regularization parameter. The optimization starts with an assumed object, simulates a set of measurements from the assumed object, compares the experimental and simulated measurements, and then updates the object based on the differences. The last step also includes the discrepancy in the prior term into the computation of the update. The process continues to iterate until a certain convergence criterion is achieved. The specific prior term $\Psi(f)$ is the key for artifact suppression and edge preservation [86, 87, 88], but such preoperative information is often difficult to acquire, or even unavailable [21]. Additionally, the value of the regularization parameter is a matter of importance yet is usually obtained by trial and error.

6.5 Deep reconstruction network with learned prior

Recently, deep learning based inversion has been proposed for X-ray tomography. The approach, known as a deep reconstruction network, utilizes a prior distribution that is either learned from a paired dataset consisting of pairs of ground truth and measurements of ground truth or the deep image prior to generate high-quality reconstructions. Here, we focus on methods with a learned prior distribution.

There are mainly two kinds of deep reconstruction networks with learned priors. The first kind is an end-to-end approach, where a direct mapping from the measurement to the object reconstruction is obtained by using measurement and ground truth object pairs as the training dataset [89]. The network implicitly learns the inverse mapping and the prior $\Psi(f)$ simultaneously. However, end-to-end deep neural networks may conflate reconstructions whose difference lies either in or close to the null space of the system matrix, leading to vulnerabilities and instabilities in the reconstruction [79, 80]. This leads to the second kind of network that removes artifacts within the FBP reconstructions [90, 91, 92, 22], in which the reconstruction is a two-step process. The first step is to use the measurement via the FBP method to produce a noisy reconstruction, and the second step is to use a deep network to remove the noise and artifacts within the FBP reconstruction. This way, the network only learns the prior $\Psi(f)$ from the FBP reconstruction and ground truth object pairs without considering the inverse mapping from measurement to object. Some works replace the FBP algorithm in the first step using another deep reconstruction network [93, 82] to overcome the issue of the instabilities.

Chapter 7

Physics-assisted Generative Adversarial Network

7.1 Physics-assisted Generative Adversarial Network

Our proposed PGAN improves upon the second kind of deep reconstruction network using the two-step reconstruction process. Rather than an FBP reconstruction, we utilize a maximum-likelihood estimation resulting from an iterative algorithm with physical priors to map the measurement to an approximant object. The physical priors that we know *a priori*, i.e., the forward operator of the X-ray tomography and the Poisson corruption in low photon measurement, are incorporated in the first step of the reconstruction. Then, a generative model uses the learned prior $\Psi(f)$ to further improve the maximum-likelihood estimate. Therefore, the PGAN inversion framework incorporates the known physical prior from an iterative algorithm and the learned prior from a deep generative model, drastically improving the reconstruction quality. The details of our algorithm are presented below.

7.2 Maximum-likelihood estimate

The maximum-likelihood estimate is the tomographic reconstruction from an iterative algorithm serving as the input to the generative model, and also the comparison baseline of the reconstruction quality. The method produces the maximum log-likelihood reconstruction $\tilde{f}$ with a given set of tomographic measurements g (in the number of photon counts for each detector pixel) assuming that the measurement noise is consistent with a Poisson process. The method was originally proposed by my collaborator Dr. Zachary Levine. The objective is to find an optimal $\tilde{f}$ given the measurements g:

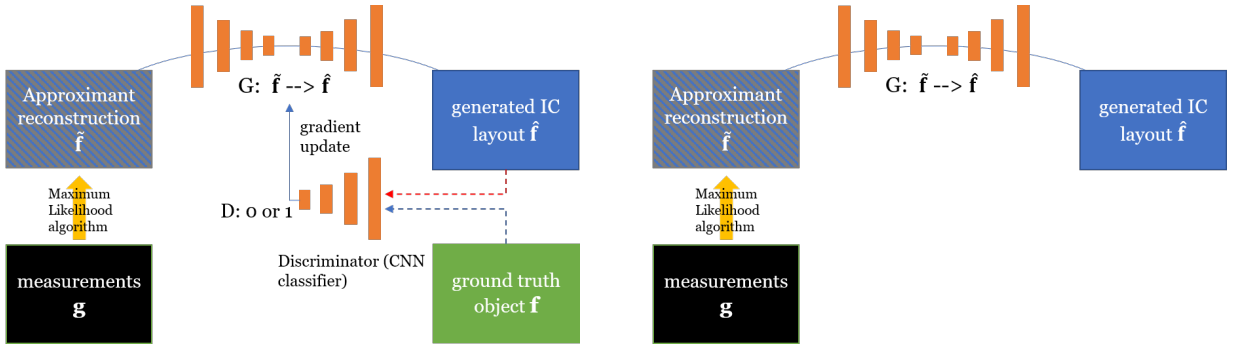
$$\tilde{f}(g) = \arg\max_{f^{(0)}} \left[L_{\mathrm{MLE}}(g|f^{(0)}) + \Psi(f^{(0)}) \right] \tag{7.1}$$

$$L_{\mathrm{MLE}}(g|f^{(0)}) = -\sum_i \left[\ln g_i! - g_i \ln g_i^{(0)} + g_i^{(0)} \right]. \tag{7.2}$$

Here L_{MLE} is the log-likelihood under the assumption of Poisson statistics [94], Ψ is a regularization function or log of the Bayesian prior, $g^{(0)}$ is the simulated measurement from a proposed object $f^{(0)}$ based on Eq. 6.2, $\sum_i$ sums over all individual measurements where i indexes the measurements at different angles, $\ln g_i!$ takes the log of the measurement at an angle and then factorizes the result element-wise, and $\tilde{f}$ is the optimal reconstruction based on maximizing the log likelihood [95, 73]. In our implementation, no Bayesian prior is included in the objective function for simplicity and clarity. The objective is maximized using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approach [96]. Discussion and results including a Bayesian prior can be found in the Appendix.

7.3 Deep generative models

Our deep generative model is based on a supervised machine learning technique known as conditional generative adversarial network (cGAN) [57]. The generative model learns a prior distribution of the object, and then improves the 3D reconstruction from maximum-likelihood estimation. When the available projection angles and photons per ray are limited for X-ray



(a) simulation results for R from 0.25, 0.5, 1, to 2 (b) runtime comparison at different R values

Figure 7-1: (a) Supervised training of the generative model using pairs $\left(\tilde{f}, g\right)$. (b) Testing on pairs never used during training.

tomography, the reconstructions from the maximum-likelihood estimation contain artifacts due to the missing cone problem. The quality of these reconstructions will eventually drop below the acceptable threshold when the angular range or photon flux of the tomographic measurements decreases. Based on our numerical experiments, the deep generative model improves the noisy maximum-likelihood reconstructions resulting in output object structure which better replicates the true object.

In GAN's original form, the objective is to map a random vector z to the targeted distribution given a set of samples from the true distribution f:

$$\arg\min_{G}\max_{D}\left(\mathbb{E}_{f}\left[\log D(f)\right]+\mathbb{E}_{z}\left[\log\left(1-D(G(z))\right)\right]\right) \tag{7.3}$$

where G is the generator and D is the discriminator. Note that the log term represents the log probability of whether the discriminator thinks the input reconstruction is realistic. The optimization process is a competition between G and D, where the generator tries to create examples as realistic as possible to deceive the discriminator while the discriminator tries to distinguish generated examples from the given true examples [97]. Therefore, the generator tries to minimize the objective function and the discriminator tries to maximize it. The cGAN method has achieved impressive results not only in computer vision [98], but also in physics-related applications, including computer-generated holography [99], medical imag-

ing [100], solving differential equations [101], and more. Conditional GAN is an extension to the original GAN model: it modifies the original GAN by conditioning both the generator and discriminator on some extra information about the distribution we try to synthesize. Whereas the original GAN has a problematic instability in training [102], cGAN gives us control over modes of the distribution to be generated.

In our case, the conditional information for the deep generative model is the noisy maximum-likelihood approximant $\tilde{f}$. This is the sole input to the generative model, and we do not include the random vector. We denote the input estimate $\tilde{f}$ as the Approximant. The training distributions are the pairs of approximant $\tilde{f}$ and ground truth f. The objective involving the discriminator of our generative model is formulated by modifying the original cGAN's objective [57], and is presented below:

$$\arg\min_{G}\max_{D}\mathbb{E}_{(f,\tilde{f})}\big[\log D(f)+\log\big(1-D(G(\tilde{f}))\big)\big] \tag{7.4}$$

The complete objective of our generative model, including the supervised term, can be found in the Appendix. Through training, the generative model learns a prior via the competition between the generator and the discriminator. When the competition reaches the Nash equilibrium, the training process completes. We denote the output of the trained generative model as $\hat{f}$. Note that the approximant $\tilde{f}$ and $\hat{f}$ are reshaped to the voxelized matrix representing the 3D object.

The generator in our model is a 3D autoencoder that first learns to convert an object representation to a latent space representation using an encoder module, and then decodes the representation back to 3D object. The discriminator is a 3D convolutional classifier that tries to find whether the output from the generator is realistic or not (reporting a floating-point number in the range of [0, 1]). Both generator and discriminator have convolutional kernels that are spectrally normalized to stabilize the training process [103]. The discriminator is updated with the generator only during training and is not needed during testing. Fig. 7-1a shows the training process for our deep generative model, and Fig. 7-1b shows the process of testing and inference. In total, four variants of the deep generative model are investigated,

in particular: deep generative model (baseline model), generative model with axial attention, generative model with a scattering representation, and generative axial model with a scattering representation. The variants alter the design of the encoding module while all sharing the same decoding modules and discriminator architecture. Detailed architectures can be found in the Appendix, and also in our github page [104].

The baseline generative model uses a series of cascaded 3D convolutional layers in alternation with pooling layers in the encoder module to extract features from the input reconstruction. Increasing the number of convolutional layers in the encoder can enable the module to learn more complicated features from the input [105], capturing high-level spatial dependencies of training objects.

Our second variant of the generative model is based on axial attention that harvests the contextual information in the input Approximant. To build such an encoder module, we replace some of the 3D convolution layers in the encoder of the generator with full axial attention modules to extract or detect global features in the input reconstructions. The core idea of this technique is to factorize the 3D self-attention into three 1D attention modules along the height, width, and depth axis sequentially, which can reduce the computational complexity of 3D self-attention to $\mathcal{O}(hwzm)$, where h, w, z, are the height, width, and depth of the input features, respectively, and m is the local constraint constant [106]. Therefore, axial attention is more efficient than the standard self-attention, enabling long-range and global feature learning to overcome the limitation of locality in the convolutional kernels. In our implementation, the local constraint is set to be the same size as the given axis, and each axial attention has eight attention heads.

The third and fourth variants of generative models include the wavelet scattering transform [107, 108] of the reconstruction as an additional input to the encoder module of the model. A scattering representation can be produced without training, and such a representation is capable of including features at multiple scales. When combined with the renormalization technique, the generative model can be further conditioned on the scattering representation in generating realistic objects. In particular, after being fed to a trainable

transformation of a fully connected layer, the wavelet representation re-scales and re-shifts the normalized feature values from convolution or axial attention. This technique may provide supplementary features of the input Approximant into the reconstruction process of the neural network [109], which may help the network in learning the mapping from noisy reconstruction to noiseless reconstruction.

Chapter 8

Evaluation methods for 3D circuits

8.1 CircuitFaker for tomographic objects

CircuitFaker is an algorithm that generates a random set of voxels with binary values resembling an integrated circuit interconnect[1]. The synthetic circuits from CircuitFaker is the class of artificial objects for tomographic reconstruction, and the implicit correlations in their spatial features are the priors to be assumed or to be learned for the inverse algorithms. A particular draw of CircuitFaker assigns a bit in each of $N = N_x N_y N_z$ locations. These locations are indexed as $i_\ell = 1, ..., N_\ell$, with $\ell = 1, 2, 3$ for x, y, and z. All bits are initialized to 0. In the first round, there are wire seed points for all locations (i_1, i_2, i_3) with i_1, ..., i_3 odd. Each seed point is set by a Bernoulli draw with probability p_w of getting a 1. There are three kinds of layers, x, y, and via layers. The x wiring layers have index $i_3 = 1 \bmod 4$. The y wiring layers have $i_3 = 3 \bmod 4$. The via layers are the others, i.e., i_3 even. In the second round, a point on an x wiring layer to the immediate right of a point with value 1 is set to 1 with probability p_x. A point on a y wiring layer immediately above in plan view a point with a value 1 is set to 1 with probability p_y. Similarly, a point on a via layer immediately above a point with a value 1 is set to 1 with probability p_z. In this thesis, we chose these parameters: $N_x = N_y = 16$, $N_z = 8$, $p_w = 0.75$, $p_x = p_y = 0.8$, and $p_z = 0.5$. Fig. 8-1 shows

[1]The CircuitFaker program was originally proposed and programmed by Dr. Zachary Levine, NIST.

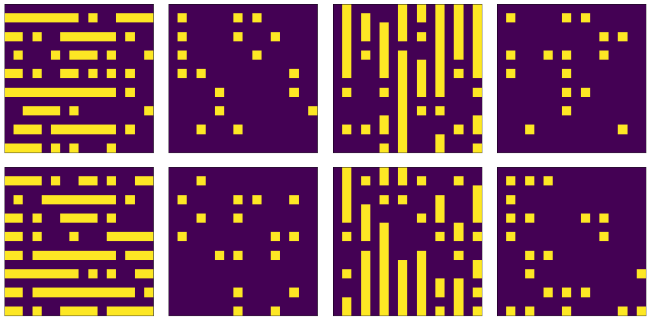


Figure 8-1: Selected $16 \times 16 \times 8$ circuit from CircuitFaker. Each image is a slice of 2D layer in the z dimension. The value of z increases as a raster scan of the 8 slices shown. Yellow indicates copper and purple indicates silicon. Here, x layers are the first (upper left) and fifth layers (lower left) in z, y layers are the third and seventh layers in z. Others are via layers.

one of the generated circuits with size $16 \times 16 \times 8$.

8.2 Imaging geometry for X-ray tomography

The imaging geometry is chosen to support an experimental project to perform integrated circuit tomography with a laboratory-scale instrument [110]. Each voxel in the circuit is of size 0.15 μm $\times$ 0.15 μm $\times$ 0.30 μm to emulate a real-world circuit. Therefore, the total volume of the circuit is 2.4 μm $\times$ 2.4 μm $\times$ 2.4 μm. The detector is assumed to be in the x-z plane at a tilt angle of $\varphi = 0°$. The rotation axis is z. The detector is 13.44 mm $\times$ 13.44 mm with 32 $\times$ 32 pixels of size 420 μm $\times$ 420 μm. The system operates with a geometric magnification of 5000, with a source-sample distance of 10 μm. There are eight tilt angles from $-30°$ to $+22.5°$ with an increment of 7.5°. There is a single source point with a cone-beam geometry. A single ray is taken from the source point to the center of each detector pixel. Minor corrections for variations in the source-to-detector pixel distance, the obliquity, and the source's Heel effect are neglected. Here, we do not use scatter corrections, and we are restricted to a single material, namely copper, at its bulk density of 8.960 g/cm^3. Therefore, the reconstruction at each voxel ends up being a binary variable. We exploit that to define the BER quality metric in the next section. The spectrum consists of two equally

weighted lines at 9362 eV and 9442 eV, the Pt Lα fluorescence lines. The attenuation per voxel is about 2 % if copper is present. The exact value depends on the details of how a ray intersects a voxel.

8.3 Bit-error-rate formulation

The bit error rate (BER) is introduced as an evaluation metric to assess the performance of the reconstruction quality. It provides a measure of the frequency of misclassification for binary values in the voxels in a given circuit. That is, BER is the probability of classifying a specific voxel in a circuit to be 1 while the ground truth value for the corresponding voxel is 0 and *vice versa*. The procedure for computing bit error rate in this thesis is slightly modified from the standard used in communication theory, and is as follows:

1. Compute posterior distributions $p(f_i = 0 \,|\, \tilde{f})$ by multiplying the probability density functions (PDFs) $p(\tilde{f} \,|\, f_i = 0)$ and $p(\tilde{f} \,|\, f_i = 1)$ and their corresponding prior distributions (p_0 and p_1 for f). Here, f_i represents an individual voxel in the circuit.

2. Apply a threshold based on a likelihood function to classify 0 and 1, where the intersection of the distributions of 0 and 1 determines the threshold. In our implementation, the prior likelihood functions are normal distributions.

3. Compute the error rates for 0 and 1 (η_0 and η_1, respectively) by summing over the misclassified region in the probability density functions.

4. Derive the expected bit error rate: $\eta_{\text{avg}} = \eta_0 p_0 + \eta_1 p_1$.

Chapter 9

Numerical results for 3D circuits

To demonstrate the ability of the learned prior from the generative approach, we investigate its performance in solving ill-conditioned tomography problems. The imaging condition is constrained to limited angle and low photon cases where the effect of ill-conditioning becomes severe. Fig. 9-1 shows selected examples of IC reconstructions at limited-angle and low photon conditions. The angular range is fixed at $-30°$ to $22.5°$ with $7.5°$ steps for all. Each row represents different reconstruction methods, and each column represents the same location at the given IC distribution with a different photon budget per ray. The last row represents the IC ground truth. Considerable improvement is visible when comparing the maximum-likelihood reconstructions and generative reconstructions.

9.1 Limited angle and low photon tomography

The quantitative comparison of limited angle and low photon tomography is shown in Fig. 9-2. The x axis is the number of photons per ray in the tomographic projection which ranges from 100 to 10^4. The y axis is the averaged bit error rate of the reconstructed 3D IC test dataset. The angular range is fixed at $-30°$ to $22.5°$ with $7.5°$ steps as well. There are five reconstructions we are comparing: reconstruction based on maximum-likelihood estimation, and reconstruction based on the four variants of generative model. For generative models, the

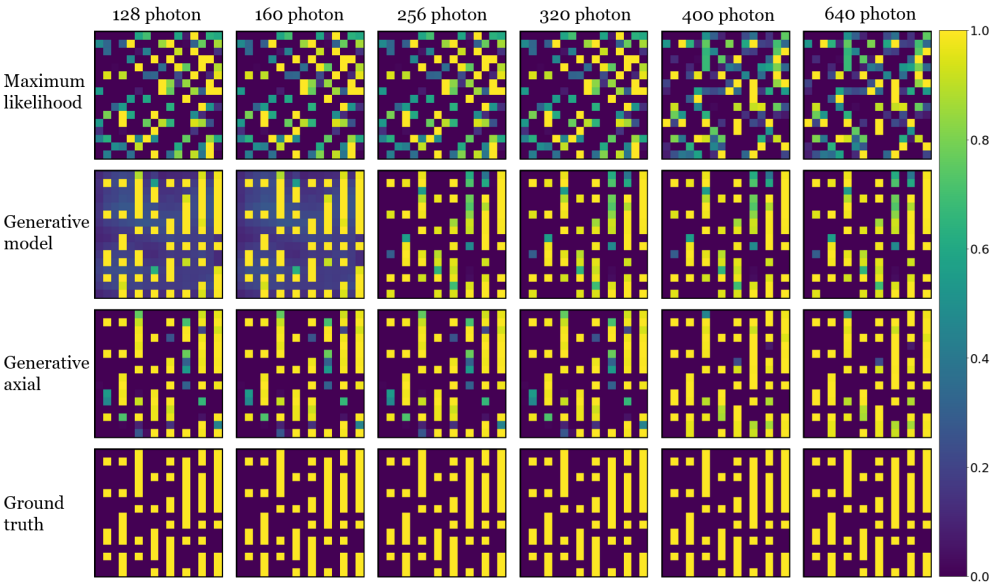


Figure 9-1: Selected examples of IC reconstructions with an angular range of $-30°$ to $22.5°$. The color scale runs from 0 to 1.

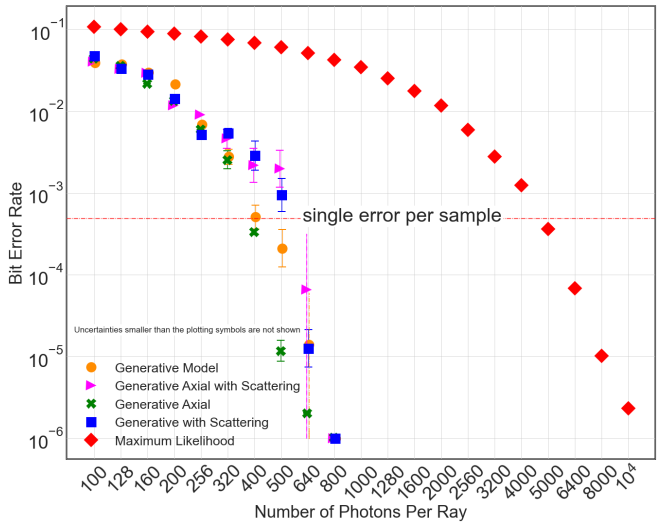
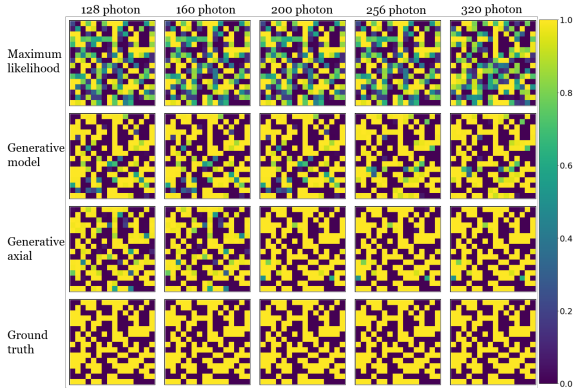


Figure 9-2: Maximum-likelihood vs. generative model reconstructions with an angular range of $-30°$ to $22.5°$.

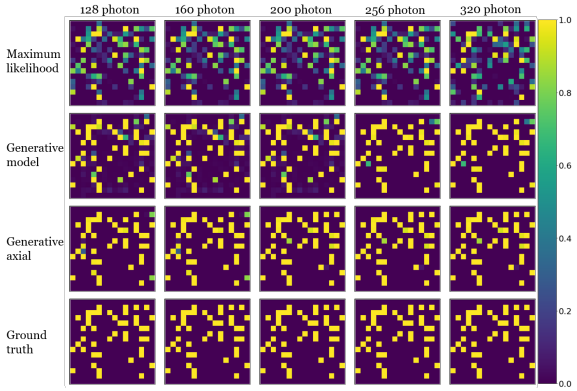
transition from above a single-error-per-sample to below happens between 320 to 640 photons per ray. Each of the simulations for the generative model is done with 1800 training sets and 200 test sets, and IC data is $16 \times 16 \times 8$ voxels. For the transition cases between 320 and 640 photons per ray, we repeat the simulation with a total of five independent synthetic sets of IC circuits and report the means and standard errors in the plot. With 640 photons per ray, the bit error rates from generative model reconstructions drop at least two orders of magnitude relative to the maximum-likelihood reconstructions. In particular, the generative model with axial attention performs the best in terms of its lower mean and standard error, reaching a single error per sample at 400 photons per ray. We may attribute this to the application of axial attention in capturing long-range interactions within the input. Generative models with wavelet scattering representation show no advantage in performance. This may be due to the small size of the input, where the additional information from the wavelet representation may have been learned from convolutional kernels and axial-attentions. The maximum-likelihood reconstructions reach a single error per sample when the number of photons per ray is around 5000. Therefore, in simulation, the generative models can reduce the photon budget to reach a single error per sample by one order of magnitude.

To confirm that the improvement from the generative approach may indeed be attributed fairly to the learned prior, we further demonstrate the quantitative comparison of limited angle and low photon tomography on 3D objects that are not spatially correlated. These 3D objects are generated with an independent coin toss at every voxel. The probability of being 1 (copper) is 0.5 for fair coin toss, or unfair coin toss that matches the fill fraction for CircuitFaker generated circuits (which is $p = 0.18521$, showing 1 standard deviation of statistical uncertainty). The learned prior in these cases is the probability p for each voxel. Since the voxels are not spatially correlated, the learned prior from the generative model is expected to be less effective in solving the inverse problem.

Fig. 9-3 shows the selected examples of independent 3D object reconstruction by limited angle and low photon tomography. The imaging geometry is the same as before, where the angular range is fixed at -30° to 22.5° with 7.5° steps. Each of the simulations for the



(a) $p = 0.5$ Bernoulli trial (fair coin)



(b) $p = 0.18521$ (unfair coin matching circuit fill factor)

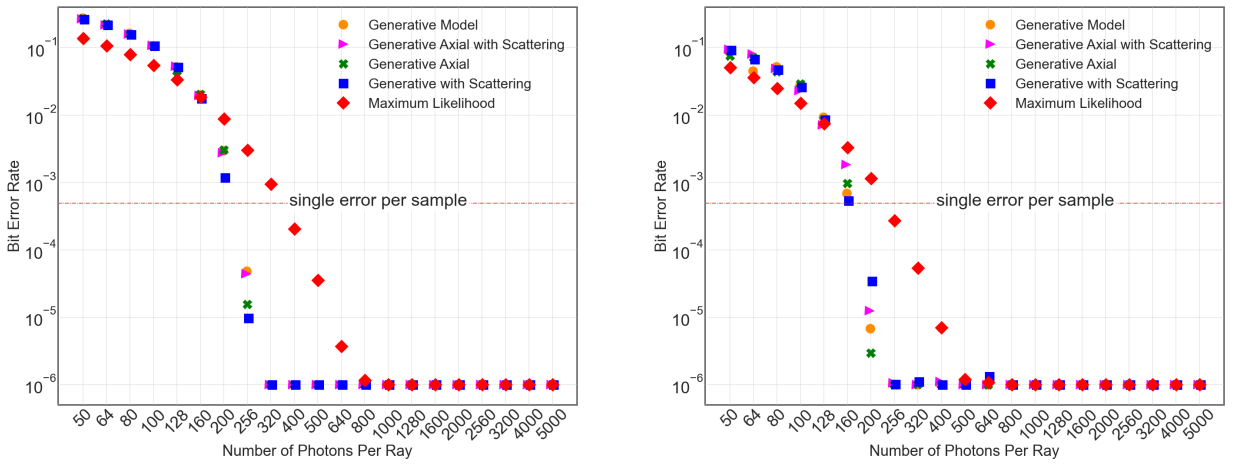
Figure 9-3: Selected examples of independent coin toss an angular range of $-30°$ to $22.5°$. The color scale runs from 0 to 1.

generative model is repeated with 1800 training sets and 200 test sets, and the independent 3D object is in $16 \times 16 \times 8$ voxels as well. The improvement from the deep generative model is less pronounced than having a circuit object.

The assumed prior (namely, Poisson noise in the measurement and that each voxel has a value in $[0, 1]$ with uniform probability) in our maximum-likelihood approach is now more proper to the reconstruction object with independent voxel. Therefore, the maximum-likelihood estimate improves. The learned prior from the deep generative model behaves similarly to a better classification cut-off for each voxel: generative models may predict each voxel value centered around 0 and 1 since all the 3D objects for training are binary. The maximum-likelihood approach does not assume objects that are binary and it may produce reconstructions with more significant variances to the mean at the same imaging condition.

9.2 Independent 3D object

Fig. 9-4 shows the quantitative comparison for independent 3D object reconstructions. The x axis is the number of photons per ray in the tomographic projection ranging from 50 to 5000, y axis is the averaged bit error rate of the reconstructed 3D coin toss test dataset. Fig. 9-4(a) is the case with fair coin toss. Compared with Fig. 9-2, the required number of photons per



(a) $p = 0.5$ Bernoulli trial (fair coin)

(b) $p = 0.18521$ (unfair coin matching circuit fill factor)

Figure 9-4: Results for independent coin toss at every voxel with an angular range of $-30°$ to $22.5°$.

ray to achieve single-error-per-sample reduced from 5000 to the range between 320 and 400 for maximum-likelihood estimation. This is attributed to a more proper prior than the maximum likelihood approach assumed, which leads to a better quality than the maximum-likelihood estimate. The generative models are slightly worse than the maximum-likelihood estimate at lower photon cases. Limited improvements are visible as the generative models need 200 to 256 photons per ray to achieve single-error-per-sample, reducing the required total number of photons for high-fidelity reconstruction. This is from the learned prior that provides a better classification cut-off, and the learned cut-off only improves the reconstruction quality when there is sufficient information for maximum-likelihood estimation. Fig. 9-4(b) is the case with a biased coin toss that has $p = 0.18521$. Compared with Fig. 9-4(a), the required number of photons per ray for maximum-likelihood estimation to achieve single-error-per-sample is slightly reduced to the range between 200 and 256. With a lower probability of having copper, the 3D objects are now more sparse. Therefore, less attenuation from the copper material leads to effectively more photons captured by the detector pixel, improving the quality of the limited-angle measurements. Also, improvement from the generative models is visible as the deep learning algorithms learn a better classification cut-off for each voxel. These results confirm that the deep learning approach benefits from the learned prior: when

the assumed prior in the iterative algorithm is not well-suited for the reconstruction object (as for the case of circuit reconstruction), the generative models can drastically improve the reconstruction quality. On the other hand, when the prior distribution itself is simple, and the assumed prior matches the distribution (for the independent coin toss object), then the generative models may only provide a marginal improvement over the iterative algorithm.

Chapter 10

Conclusion

We have demonstrated a reliable physics-informed machine learning-based computational imaging method that works well for Randomized Probe Imaging with 2D phase-only objects. This method has been further extended to 3D tomographic reconstruction by replacing the 2D convolution to 3D convolution. The fully trained machine learning method is, as expected, more computationally efficient and produces higher fidelity reconstructions. Compared with iterative algorithm, physics-informed machine learning can reduce the photon requirement to achieve a given error rate. We further attribute the improvement to the learned prior by reconstructing objects created without spatial correlations. The improved resilience to noise makes our approach attractive in situations where illumination power is limited or the samples are sensitive to excessive radiation exposure for 2D and 3D X-ray imaging.

Appendix A

Discussion of End-to-End RPI phase retrieval

Let $\Gamma \doteq \{O_i, I_i\}$ be the paired training dataset, P the known randomized probe, and let $G_{\mathbf{w}}$ be a set of parameters for the deep neural network that can be trained with. The parameters $\mathbf{w}$ are also commonly referred to as the "connection weights" or simply "weights" in traditional neural network architectures. Then the end-to-end phase retrieval problem becomes that of finding the optimal weights $\hat{\mathbf{w}}$ such that given any intensity pattern within the dataset distribution Γ, along with the known probe P, the network can produce a generated object O_i^+ that is an equivalent class to the O_i, or formally

$$\hat{\mathbf{w}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_i \mathscr{L}\big\{O_i, G_{\mathbf{w}}(I_i, P)\big\} \tag{A.1}$$

where $\mathscr{L}$ is the loss function that measures the discrepancy between the generated object O_i^+ and ground truth O_i. For RPI, the equivalence class is defined to be the set of all objects which may be derived from the $E_i(x, y)$ by changing in the global phase. The commonly encountered spatial shift and time-reversal symmetries in diffractive imaging systems are not symmetries of the RPI system, due to the presence of the randomized probe [39]. For global phase degeneracy, any complex rotation of O_i^+ in degree ϕ would result in identical far-field

intensity pattern I_i, and therefore, the output O_i^+ of the formulation above also needs to take those degenerate solutions into account. An alternative formulation inspired by [111] would be

$$\hat{\mathbf{w}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_i \mathscr{L}\left\{I_i, |\mathscr{F}\{G_{\mathbf{w}}(I_i, P)\}|^2\right\} \tag{A.2}$$

Here, the problem of phase retrieval becomes equivalent to that of minimizing the loss in the far-field domain, *i.e.,* the spatial frequency domain. Thus, the inverse problem is indirectly solved, with the optimization forcing the network to generate the amplitude and phase of the exiting wave E, rather than the object O. Since the applied constraint is in the far-field domain, the formulation would preserve the global phase degeneracy in its solution. However, in this case, the network would learn priors based on the training distribution E, and it would be challenging to continuously sample this distribution and capture its statistics for testing as E is the product of the object and randomized probe. It is easier to guarantee that the training distributions O follow the same statistics of the testing distribution O, as long as training and testing datasets are both constrained to natural images with geometric features.

Appendix B

Network training procedure

Our proposed deep k-learning networks were implemented in Python 3.7.9 using TensorFlow 2.3.1, and trained with NVidia V100 tensor core graphics processing unit on MIT Supercloud [112]. The object training set was from 4,000 natural images in ImageNet, where phases were set to be the images and amplitudes were set to be one. The $(256 \times 256 \times 3)$ ImageNet images were converted to gray-scale from the original RGB format. Therefore, the total training object dataset is a complex matrix with dimension of $(4000, 256, 256, 1)$. The randomized probe P was generated based on the method in [113] given the sampling ratio R. The far-field diffraction patterns were then numerically simulated based on the optical setup in Figure 2-1. The approximate objects were subsequently generated via automatic differentiation with one iteration with steepest gradient descent for each diffraction pattern, and the loss function $\mathscr{L}$ here is the mean square error (MSE) on the amplitude. The iterative results are from 100 iterations with 0.5 learning rate. After numerical simulation, we normalized all of the paired training data in the $\Gamma \doteq \{O_i, O_i^*, I_i\}$ dataset between $[0, 255]$. This will be shown later to improve network training stability. For training, Adam optimizer [114] was used with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, the initial learning rate was 2×10^{-4}. The validation split was 0.1 to provide an unbiased evaluation of a model fit on the training dataset.. The learning rate would be reduced by half when the validation loss stops improving for 10 epochs. We set the maximum epoch to be 200, and the training will

stop early when either the validation loss plateaus for 20 epochs or the minimal learning rate 10^{-8} is reached. This early-stop technique prevents the model from over-fitting. We keep the same training parameters for all the networks, the variations of different training were i), the training strategy (either end-to-end or deep-k-learning), ii), generative or non generative, iii), network weights initialization (with random initial weights or ImageNet pre-trained weights in the encoder arm), and iv) hyper-parameter β for generative deep k-learning (α is fixed as $1/8$ to reduce the complexity of hyper-parameter grid search) in the total loss function of the autoencoder/generator. When the network is initialized with pre-trained weights in the encoder, the 200 epochs are completed in two steps: in the first step, we only train the decoder of the network while the encoder is frozen with pre-trained weights; in the second step, we unfreeze and train the entire network. This can accelerate the training for models with pre-trained weights.

For end-to-end training, we divided each far-field diffraction pattern into multiple patches with dimension of $(256, 256, C_R)$, where C_R is the number of channels that depends on the dimension of the diffraction pattern with the given oversampling ratio R. The inputs to the end-to-end network are the multi-patch representation of diffraction pattern concatenated with the randomized probe that is also in multi-patch representation. This way, we can keep the number of parameters in the end-to-end network to be roughly the same as in the deep-k-learning framework (around 76.5 million in total parameters in both cases, not counting the discriminator network and pre-trained EfficientNetB0), and makes the subsequent performance comparison fair. Also, in the end-to-end neural network, we removed the skip connections between encoder and decoder because of the large domain transfer in-between.

Appendix C

Experimental procedure for RPI measurements

Table C.1: Summary of the four sets of experimental measurements

Target photon/pixel	**Measured photon/pixel**	**Averaged SNR**
1000	996	6.09
100	127	2.07
10	11.9	0.375
1	1.77	0.0525

Data were collected under four different experimental imaging conditions individually. We thank Abe Levitan for experimental collaboration. For the 10, 100, and 1000 photon per object pixel collections, the total image intensity was modulated by extending the exposure time, using an EM gain of 54 (corresponding to EM level of 3800 in the camera software) and an offset level of 0. To implement the necessary range of attenuations, we chose a pinhole size of 5μm, significantly smaller than the waist of the beam emerging from the collimating objective; and moved the pinhole away from the center to further lower photon fluxes. The offset level for this measurement was set to 500, due to the extremely weak signal level. We also collected 10 background images per signal level under a reproduction of the imaging conditions, with the laser turned off.

The number of photons per object pixel was calculated empirically by summing over the captured diffraction signal in each image, with the mean background signal for that imaging condition subtracted off. After multiplying by a previously calibrated conversion factor to convert between ADUs and photon counts [115], we were able to calculate the mean number of photons measured on the detector under the respective imaging condition.

To calculate the reported signal to noise ratios, we separated the noise contribution into signal-dependent and signal-independent contributions. The signal-independent portion, which included readout noise, dark current, and shot noise from background photons, was calibrated empirically using the statistics of the dark images. Specifically, the standard deviation of the background images was calculated in binned 8 by 8 pixel regions to produce a low-resolution map of the empirical signal-independent noise level. We estimated the signal-dependent contribution by assuming it is dominated by Poisson noise. Under this assumption, the standard deviation of the signal-dependent noise can be estimated by the square root of the measured signal (minus the mean background) at each pixel. The total variance at each pixel is thus determined by the sum of the squares of the standard deviations of the two contributions. The reported signal to noise ratios are defined as the ratio of the sum of the signal image (the total power in the signal channel across the entire image) to the sum of the calculated standard deviations due to noise (the total power in the noise channel across the image).

At each photon incidence rate condition, we first took a 31×31 step ptychography dataset with 75μm steps in order to retrieve the probe and background states. Scanning for the ptychography dataset was implemented by shifting a displayed image digitally across the SLM. Ptychographic reconstructions were performed via automatic differentiation ptychography using the Adam algorithm, with a single probe mode and a quadratic background correction. A learning rate scheduler was used to lower the learning rate by a factor of 0.2 at plateaus to ensure good convergence. After performing the reconstruction we displayed the ImageNet images, upsampled so that each pixel in the image covered a 2 by 2 pixel region on the SLM, in series to collect the RPI datasets.

Appendix D

Maximum-likelihood estimation with a Bayesian prior

With classical regularization, a proper Bayesian prior and its regularization weight are usually not straightforward to choose for a given object distribution. We include an example using a Bouman-Sauer prior to demonstrate this challenge, and also to support our choice of not including the Bayesian prior in our maximum-likelihood estimation. We thank Dr. Zachary Levine here for performing the simulations.

The selected Bouman-Sauer prior imposes the smoothness of the reconstruction. It is very similar to the Total Variation (TV) prior, except Bouman-Sauer is more general [116]. Fig. D-1 shows the reconstruction quality versus the regularization weight for three imaging conditions with limited angles. The measurements are the 200 testing dataset with 3D IC as the tomographic object, identical to dataset that generates Fig. 9-2. Note that when the weight is 0, the reconstruction is identical to the maximum-likelihood estimate without the prior (our baseline in the main text). The Bouman-Sauer prior provides limited improvement for 256 photons per ray case. Quality degradation is obvious for the cases of 1000 and 4000 photons per ray.

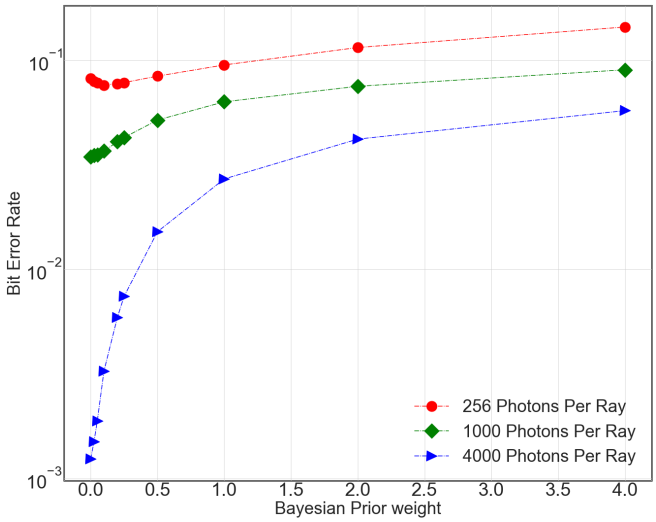


Figure D-1: Maximum-likelihood reconstructions including the Bouman-Sauer prior with an angular range of $-30°$ to $22.5°$.

Appendix E

Network details for PGAN

E.1 Network architecture

The general description of the network architecture is given in Section 7.3. The code to generate the networks is publicly available on github [104]. Here, we include more information about the network architecture for reproducibility.

Fig. E-1 is the detailed network architecture for the deep generative model (the generator). The overall design is based on UNet [117] to perform pixel-by-pixel prediction (for 3D reconstruction, where the 3D object is voxelized by a 3D matrix/array). The input dimension to the model is in $(16, 16, 8, 1)$. Four DownResBlocks encode the input Approximant and produce a latent representation that is in dimension of $(1, 1, 8, 512)$. Four UpResBlocks decode the latent representation to a vector in dimension of $(16, 16, 8, 64)$. Concatenated skip-connections are used in between the last three DownResBlocks and the first three UpResBlocks to preserve high frequency information of the input Approximant [46]. Dropout layers are included to prevent over-fitting. The final layer of convolution reduces this vector to a final output in $(16, 16, 8, 1)$, and a Tanh layer forces the final output to the range between -1 and 1. The DownResBlock and UpResBlock share similar topology to the Resblock in ResNet [118], except the use of different 3D sampling layers. Here, we implemented downsampling and upsampling layers that only sample the dimension in height and width

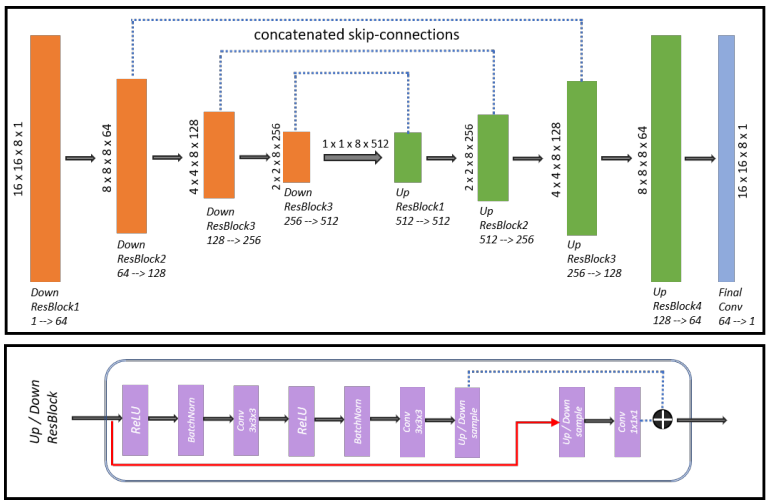


Figure E-1: Network architecture for the deep generative model (generator)

but not depth.

For the base generative model, feature extraction in the DownResBlock and UpResBlock is achieved by 3D convolutional kernel with spectral normalization [103]. For the axial-attention based model, feature extraction in the DownResBlock is achieved by the mixture of 3D convolutional kernel and axial attention both with spectral normalization.

For models including the wavelet scattering transform, the wavelet representation for the input Approximant is first produced by HarmonicScattering3D in the Kymatio package [119] with $J = 2$ (maximum scale of 2^2), integral powers with $\{0.5, 1.0, 2.0, 3.0\}$. Then, the batch normalization layers in the UpResBlock are replaced by conditional batch normalization (CBN) layers [120, 121, 122], where the conditional information is the wavelet representation. Note that the fully connected layers within the CBN are spectrally normalized as well.

The discriminator for all the generative models is the same, with four DownResBlocks bringing the input from dimension $(16, 16, 8)$ to $(2, 2, 1024)$, following with a reduce sum operation to bring it further to a vector of $(1, 1, 1024)$. A fully connected layer followed thereafter to produce a floating point number for classification.

E.2 Network training

Our proposed networks are implemented in Python 3.7.9 using TensorFlow 2.3.1, and trained with an NVIDIA V100 tensor core graphics processing unit on MIT Supercloud [112]. An Adam optimizer [114] is used with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The two time-scale update rule (TTUR) is used to stabilize the training of the generative network [123, 124], where the initial learning rate is 10^{-4} for the generator and 4×10^{-4} for the discriminator. In each iteration, the generator is updated four times while the discriminator is updated once.

Training sets of 1800 reconstructions are generated independently for each condition studied, except that the ground truth is common. The batch size for training is 20. An additional 200 reconstructions per condition are used for testing. The learning rate is reduced by half when the validation loss stops improving for 5 iterations. We set the maximum number of iterations to be 200, and the training stops early when either the validation loss plateaus for 20 iterations, or the minimum learning rate 10^{-8} is reached. This early-stop technique can prevent the model from over-fitting. The loss function for the autoencoder/generator consists of two parts: supervised loss and adversarial loss. We choose supervised loss to be the negative of the Pearson correlation coefficient $r_{f,\tilde{f}}$, which is defined as

$$r_{f,\tilde{f}} = \frac{\mathrm{cov}(f, \tilde{f})}{\sigma_f \, \sigma_{\tilde{f}}}, \tag{E.1}$$

where cov is the covariance and σ is the standard deviation. The total objective of training is to find the optimal generator G_{opt} given the Approximant $\tilde{f}$ and ground truth f:

$$G_{\mathrm{opt}}(\tilde{f}) = \arg\min_G \max_D \mathbb{E}_{(f,\tilde{f})}\big\{ - r_{f,G(\tilde{f})} + \lambda\big[\log D(f) + \log\big(1 - D(G(\tilde{f}))\big)\big]\big\} \tag{E.2}$$

The hyper-parameter λ controls the degree of generation from input noise to features. In our experiments, λ ranges from $1/2^0$ to $1/2^6$ with an incremental factor of $1/2$. The loss

function for GAN is the hinge loss [125], and is defined below:

$$\begin{aligned} L_D &= \text{mean}\big\{\min\{0, 1 - D(f)\}\big\} + \text{mean}\big\{\min\{0, 1 + D(G(\hat{f}))\}\big\} \\ L_G &= -\text{mean}\{D(G(\hat{f}))\} \end{aligned} \tag{E.3}$$

Here, L_G is the loss for generator and L_D is the loss for discriminator. The operator min(...) chooses the smaller value between the two inputs. The mean is taken over the batch of the training data.

Appendix F

Convergence and stability of the deep generative network

As mentioned in Section 7.3, when first proposed, GANs had an instability problem during training. The model was easy to collapse, generating non-satisfactory results [126]. It was since the appearance of deep convolutional generative adversarial networks (DCGAN) [123] that researchers began making GANs more stable by improving the structure and training skills. Later, the Wasserstein Generative Adversarial Network (WGAN) was introduced and provided a more detailed explanation of GANs' poor control [59]. A solution was also proposed, i.e., imposing Lipschitz continuity, to improve the quality of generated results [59, 127]. Nowadays, there are well-known techniques to overcome the challenge in training GAN. We summarized the techniques we used in PGAN for interested researchers.

F.1 Spectral normalization

While the WGAN approaches impose the Lipschitz continuity by gradient clipping or gradient penalty to stabilize the training, spectral normalization imposes a similar constraint by normalizing the weights within the network. This normalization technique is computationally light and easy to incorporate into existing implementations, and has been shown

effective in many applications [128, 129, 130].

F.2 Hinge loss

Hinge loss has shown improved performance when combined with spectral normalization. Therefore, it has become standard in recent state of the art GANs [131].

F.3 Two time-scale update rule (TTUR)

TTUR provides theoretical convergence of the GAN to a stationary local Nash equilibrium [124]. The core idea is to have an individual learning rate for both the discriminator and the generator. In our implementation, we choose 4×10^{-4} for the discriminator and 10^{-4} for the generator.

Bibliography

[1] Z. Guo, A. Levitan, G. Barbastathis, and R. Comin, "Randomized probe imaging through deep k-learning," in *OSA Imaging and Applied Optics Congress 2021 (3D, COSI, DH, ISA, pcAOP)*, p. CTh7A.6, Optical Society of America, 2021.

[2] Z. Guo, A. Levitan, G. Barbastathis, and R. Comin, "Randomized probe imaging through deep k-learning," *Opt. Express*, vol. 30, pp. 2247–2264, Jan 2022.

[3] I. Peterson, B. Abbey, C. Putkunz, D. Vine, G. van Riessen, G. Cadenazzi, E. Balaur, R. Ryan, H. Quiney, I. McNulty, A. Peele, and K. Nugent, "Nanoscale fresnel coherent diffraction imaging tomography using ptychography," *Opt. Express*, vol. 20, pp. 24678–24685, Oct 2012.

[4] H. N. Chapman and K. A. Nugent, "Coherent lensless x-ray imaging," *Nature Photonics*, vol. 4, no. 12, pp. 833–839, 2010.

[5] M. Holler, M. Odstrcil, M. Guizar-Sicairos, M. Lebugle, E. Müller, S. Finizio, G. Tinti, C. David, J. Zusman, W. Unglaub, O. Bunk, J. Raabe, A. F. J. Levi, and G. Aeppli, "Three-dimensional imaging of integrated circuits with macro- to nanoscale zoom," *Nature Electronics*, vol. 2, pp. 464–470, Oct 2019.

[6] C. Y. Hémonnot and S. Köster, "Imaging of biological materials and cells by X-ray scattering and diffraction," *ACS Nano*, vol. 11, no. 9, pp. 8542–8559, 2017.

[7] C. Muehleman, J. Li, D. Connor, C. Parham, E. Pisano, and Z. Zhong, "Diffraction-enhanced imaging of musculoskeletal tissues using a conventional X-ray tube," *Academic Radiology*, vol. 16, no. 8, pp. 918–923, 2009.

[8] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied Optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[9] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[10] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: a contemporary overview," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[11] R. L. Sandberg, A. Paul, D. A. Raymondson, S. Hädrich, D. M. Gaudiosi, J. Holtsnider, R. I. Tobey, O. Cohen, M. M. Murnane, H. C. Kapteyn, C. Song, J. Miao, Y. Liu, and F. Salmassi, "Lensless diffractive imaging using tabletop coherent high-harmonic soft-x-ray beams," *Phys. Rev. Lett.*, vol. 99, p. 098103, Aug 2007.

[12] A. Tripathi, I. McNulty, and O. G. Shpyrko, "Ptychographic overlap constraint errors and the limits of their numerical recovery using conjugate gradient descent methods," *Optics Express*, vol. 22, no. 2, pp. 1452–1466, 2014.

[13] P. Sidorenko and O. Cohen, "Single-shot ptychography," *Optica*, vol. 3, no. 1, pp. 9–14, 2016.

[14] B. Lee, J.-y. Hong, D. Yoo, J. Cho, Y. Jeong, S. Moon, and B. Lee, "Single-shot phase retrieval via fourier ptychographic microscopy," *Optica*, vol. 5, no. 8, pp. 976–983, 2018.

[15] D. Goldberger, J. Barolak, C. G. Durfee, and D. E. Adams, "Three-dimensional single-shot ptychography," *Optics Express*, vol. 28, no. 13, pp. 18887–18898, 2020.

[16] A. L. Levitan, K. Keskinbora, U. T. Sanli, M. Weigand, and R. Comin, "Single-frame far-field diffractive imaging with randomized illumination," *Optics Express*, vol. 28, no. 25, pp. 37103–37117, 2020.

[17] Z. Liu, T. Bicer, R. Kettimuthu, D. Gursoy, F. De Carlo, and I. Foster, "Tomogan: low-dose synchrotron X-ray tomography with generative adversarial networks: discussion," *JOSA A*, vol. 37, no. 3, pp. 422–434, 2020.

[18] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, "Deep-learning tomography," *The Leading Edge*, vol. 37, no. 1, pp. 58–66, 2018.

[19] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier, "Deep learning computed tomography," in *International conference on medical image computing and computer-assisted intervention*, pp. 432–440, Springer, 2016.

[20] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nature Medicine*, vol. 25, pp. 954–961, Jun 2019.

[21] H. Zhang, L. Li, K. Qiao, L. Wang, B. Yan, L. Li, and G. Hu, "Image prediction for limited-angle tomography via deep learning with convolutional neural network," *arXiv preprint arXiv:1607.08707*, 2016.

[22] I. Kang, A. Goy, and G. Barbastathis, "Dynamical machine learning volumetric reconstruction of objects' interiors from limited angular views," *Light Sci. Appl.*, vol. 10, 12 2021.

[23] A. Goy, G. Rughoobur, S. Li, K. Arthur, A. I. Akinwande, and G. Barbastathis, "High-resolution limited-angle phase tomography of dense layered objects using deep neural networks," *Proceedings of the National Academy of Sciences*, vol. 116, no. 40, pp. 19848–19856, 2019.

[24] Z. Guan and E. H. Tsai, "Ptychonet: Fast and high quality phase retrieval for ptychography," tech. rep., Brookhaven National Lab.(BNL), Upton, NY (United States), 2019.

[25] T. Nguyen, Y. Xue, Y. Li, L. Tian, and G. Nehmetallah, "Deep learning approach for Fourier ptychography microscopy," *Optics Express*, vol. 26, no. 20, pp. 26470–26484, 2018.

[26] Y. Chen, Z. Luo, X. Wu, H. Yang, and B. Huang, "U-net CNN based Fourier ptychography," *arXiv preprint arXiv:2003.07460*, 2020.

[27] L. Boominathan, M. Maniparambil, H. Gupta, R. Baburajan, and K. Mitra, "Phase retrieval for Fourier ptychography under varying amount of measurements," *arXiv preprint arXiv:1805.03593*, 2018.

[28] J. Zhang, T. Xu, Z. Shen, Y. Qiao, and Y. Zhang, "Fourier ptychographic microscopy reconstruction with multiscale deep residual network," *Optics Express*, vol. 27, no. 6, pp. 8612–8625, 2019.

[29] Y. Rivenson, Y. Wu, and A. Ozcan, "Deep learning in holography and coherent imaging," *Light: Science & Applications*, vol. 8, no. 1, pp. 1–8, 2019.

[30] R. Horisaki, R. Takagi, and J. Tanida, "Deep-learning-generated holography," *Applied Optics*, vol. 57, no. 14, pp. 3859–3863, 2018.

[31] M. H. Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, "DeepCGH: 3D computer-generated holography using deep learning," *Optics Express*, vol. 28, no. 18, pp. 26636–26650, 2020.

[32] Z. Ren, H. K.-H. So, and E. Y. Lam, "Fringe pattern improvement and super-resolution using deep learning in digital holography," *IEEE Transactions on industrial informatics*, vol. 15, no. 11, pp. 6179–6186, 2019.

[33] A. Goy, K. Arthur, S. Li, and G. Barbastathis, "Low photon count phase retrieval using deep learning," *Physical Review Letters*, vol. 121, no. 24, p. 243902, 2018.

[34] M. Deng, S. Li, A. Goy, I. Kang, and G. Barbastathis, "Learning to synthesize: Robust phase retrieval at low photon counts," *Light: Science & Applications*, vol. 9, no. 1, pp. 1–16, 2020.

[35] I. Kang, F. Zhang, and G. Barbastathis, "Phase extraction neural network (PhENN) with coherent modulation imaging (CMI) for phase retrieval at low photon counts," *Optics Express*, vol. 28, no. 15, pp. 21578–21600, 2020.

[36] Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," *Light: Science & Applications*, vol. 7, pp. 17141–17141, Oct. 2017.

[37] Y. Xue, S. Cheng, Y. Li, and L. Tian, "Reliable deep-learning-based phase imaging with uncertainty quantification," *Optica*, vol. 6, no. 5, pp. 618–629, 2019.

[38] A. Matlock and L. Tian, "Physical model simulator-trained neural network for computational 3d phase imaging of multiple-scattering samples," *arXiv preprint arXiv:2103.15795*, 2021.

[39] A. Fannjiang and W. Liao, "Phase retrieval with random phase illumination," *JOSA A*, vol. 29, no. 9, pp. 1847–1859, 2012.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[42] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[43] C. Metzler, P. Schniter, A. Veeraraghavan, and richard baraniuk, "prDeep: Robust phase retrieval with a flexible deep network," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 3501–3510, PMLR, 10–15 Jul 2018.

[44] Y. Zhang, M. A. Noack, P. Vagovic, K. Fezzaa, F. Garcia-Moreno, T. Ritschel, and P. Villanueva-Perez, "Phasegan: A deep-learning phase-retrieval approach for unpaired datasets," *arXiv preprint arXiv:2011.08660*, 2020.

[45] A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," *Optica*, vol. 4, no. 9, pp. 1117–1125, 2017.

[46] M. Deng, S. Li, Z. Zhang, I. Kang, N. X. Fang, and G. Barbastathis, "On the interplay between physical and content priors in deep learning for computational imaging," *Optics Express*, vol. 28, no. 16, pp. 24152–24170, 2020.

[47] G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica*, vol. 6, no. 8, pp. 921–943, 2019.

[48] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval.," in *ESANN*, vol. 1, p. 2, Citeseer, 2011.

[49] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

[50] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[52] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[55] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[56] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.

[57] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[58] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[59] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.

[60] S. Wang, J. Gelb, S. Lau, and W. Yun, "Metrology of 3D IC with X-ray microscopy and nano-scale X-ray CT," in *2009 IEEE International Interconnect Technology Conference*, pp. 131–133, IEEE, 2009.

[61] S. Bord, A. Clement, J. Lecomte, and J. Marmeggi, "An X-ray tomography facility for IC industry at ST Microelectronics Grenoble," *Microelectronic Engineering*, vol. 61, pp. 1069–1075, 2002.

[62] K. Mahmood, P. L. Carmona, S. Shahbazmohamadi, F. Pla, and B. Javidi, "Real-time automated counterfeit integrated circuit detection using X-ray microscopy," *Applied Optics*, vol. 54, no. 13, pp. D25–D32, 2015.

[63] A. Momose, T. Takeda, Y. Itai, and K. Hirano, "Phase-contrast X-ray computed tomography for observing biological soft tissues," *Nature Medicine*, vol. 2, no. 4, pp. 473–475, 1996.

[64] L. Salvo, M. Suéry, A. Marmottant, N. Limodin, and D. Bernard, "3D imaging in material science: Application of X-ray tomography," *Comptes Rendus Physique*, vol. 11, no. 9-10, pp. 641–649, 2010.

[65] M. Alam, H. Shen, N. Asadizanjani, M. Tehranipoor, and D. Forte, "Impact of X-ray tomography on the reliability of integrated circuits," *IEEE Transactions on Device and Materials Reliability*, vol. 17, no. 1, pp. 59–68, 2017.

[66] P. J. Withers, C. Bouman, S. Carmignato, V. Cnudde, D. Grimaldi, C. K. Hagen, E. Maire, M. Manley, A. Du Plessis, and S. R. Stock, "X-ray computed tomography," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–21, 2021.

[67] E. Kobler, M. Muckley, B. Chen, F. Knoll, K. Hammernik, T. Pock, D. Sodickson, and R. Otazo, "Variational deep learning for low-dose computed tomography," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6687–6691, IEEE, 2018.

[68] J. Wang, J. Liang, J. Cheng, Y. Guo, and L. Zeng, "Deep learning based image reconstruction algorithm for limited-angle translational computed tomography," *PLOS One*, vol. 15, no. 1, p. e0226963, 2020.

[69] Y. Huang, S. Wang, Y. Guan, and A. Maier, "Limited angle tomography for transmission X-ray microscopy using deep learning," *Journal of Synchrotron Radiation*, vol. 27, no. 2, pp. 477–485, 2020.

[70] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan, "Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography," *Inverse Problems*, vol. 35, no. 6, p. 064002, 2019.

[71] Y. Huang, A. Preuhs, G. Lauritsch, M. Manhart, X. Huang, and A. Maier, "Data consistent artifact reduction for limited angle tomography with deep learning prior," in *International Workshop on Machine Learning for Medical Image Reconstruction*, pp. 101–112, Springer, 2019.

[72] J. Frikel, *Reconstructions in limited angle X-ray tomography: Characterization of classical reconstructions and adapted curvelet sparse regularization.* PhD thesis, Technische Universität München, 2013.

[73] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Transactions on Image Processing*, vol. 2, pp. 296–310, 1993.

[74] T. Sato, S. J. Norton, M. Linzer, O. Ikeda, and M. Hirama, "Tomographic image reconstruction from limited projections using iterative revisions in image and transform spaces," *Applied Optics*, vol. 20, no. 3, pp. 395–399, 1981.

[75] D. Verhoeven, "Limited-data computed tomography algorithms for the physical sciences," *Applied Optics*, vol. 32, no. 20, pp. 3736–3754, 1993.

[76] A. Allag, R. Drai, A. Benammar, and T. Boutkedjirt, "X-rays tomographic reconstruction images using proximal methods based on L1 norm and TV regularization," *Procedia Computer Science*, vol. 127, pp. 236–245, 2018.

[77] D. Kazantsev, G. Van Eyndhoven, W. Lionheart, P. Withers, K. Dobson, S. McDonald, R. Atwood, and P. Lee, "Employing temporal self-similarity across the entire time domain in computed tomography reconstruction," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2043, p. 20140389, 2015.

[78] W. Zhang, N. Liang, Z. Wang, A. Cai, L. Wang, C. Tang, Z. Zheng, L. Li, B. Yan, and G. Hu, "Multi-energy CT reconstruction using tensor nonlocal similarity and spatial sparsity regularization," *Quantitative Imaging in Medicine and Surgery*, vol. 10, no. 10, p. 1940, 2020.

[79] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of ai," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30088–30095, 2020.

[80] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen, "The troublesome kernel: why deep learning for inverse problems is typically unstable," *arXiv:2001.01258*, 2020.

[81] J. Schwab, S. Antholzer, and M. Haltmeier, "Deep null space learning for inverse problems: convergence analysis and rates," *Inverse Problems*, vol. 35, no. 2, p. 025008, 2019.

[82] W. Wu, D. Hu, W. Cong, H. Shan, S. Wang, C. Niu, P. Yan, H. Yu, V. Vardhanabhuti, and G. Wang, "Stabilizing deep tomographic reconstruction networks," *arXiv:2008.01846*, 2020.

[83] A. Tarantola, *Inverse problem theory and methods for model parameter estimation.* SIAM, 2005.

[84] T. Kailath, "Lectures on Wiener and Kalman filtering," in *Lectures on Wiener and Kalman Filtering*, pp. 1–143, Springer, 1981.

[85] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 185–194, 1999.

[86] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing (PICCS): a method to accurately reconstruct dynamic CT images from highly undersampled projection data sets," *Medical Physics*, vol. 35, no. 2, pp. 660–663, 2008.

[87] T. Heußer, M. Brehm, L. Ritschl, S. Sawall, and M. Kachelrieß, "Prior-based artifact correction (PBAC) in computed tomography," *Medical Physics*, vol. 41, no. 2, p. 021906, 2014.

[88] M. J. Schrapp and G. T. Herman, "Data fusion in X-ray computed tomography using a superiorization approach," *Review of Scientific Instruments*, vol. 85, no. 5, p. 053701, 2014.

[89] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018.

[90] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2524–2535, 2017.

[91] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[92] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.

[93] J. He, Y. Wang, and J. Ma, "Radon inversion via deep learning," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 2076–2087, 2020.

[94] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 534–548, 1993.

[95] Z. H. Levine, T. J. Blattner, A. P. Peskin, and A. L. Pintar, "Scatter corrections in X-ray computed tomography: A physics-based analysis," *J. Res. of the Natl. Inst. of Stand. and Tech.*, vol. 124, p. 124013, 2019.

[96] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[97] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 1–9, 2014.

[98] C. Yangjie, J. Lili, C. Yongxia, L. Nan, and L. Xuexiang, "Review of computer vision based on generative adversarial networks," *Journal of Image and Graphics*, vol. 23, no. 10, pp. 1433–1449, 2018.

[99] M. H. Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, "Deepcgh: 3d computer-generated holography using deep learning," *Opt. Express*, vol. 28, pp. 26636–26650, Aug 2020.

[100] X. Yi, E. Walia, and P. Babyn, "Generative adversarial network in medical imaging: A review," *Medical Image Analysis*, vol. 58, p. 101552, 2019.

[101] S. Malik, U. Anwar, A. Ahmed, and A. Aghasi, "Learning to solve differential equations across initial conditions," in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[102] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?," in *International conference on machine learning*, pp. 3481–3490, PMLR, 2018.

[103] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv:1802.05957*, 2018.

[104] Z. Guo, "Physics-assisted generative adversarial network for X-ray tomography." https://github.com/zguo0525/Physics-assisted-Generative-Adversarial-Network-for-X-Ray-Tomography, 2021.

[105] L. J. Ba and R. Caruana, "Do deep nets really need to be deep?," *arXiv:1312.6184*, 2013.

[106] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation," in *European Conference on Computer Vision*, pp. 108–126, Springer, 2020.

[107] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, 2014.

[108] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[109] J.-B. Delbrouck and S. Dupont, "Modulating and attending the source image during encoding improves multimodal translation," *arXiv:1712.03449*, 2017.

[110] P. Szypryt, D. A. Bennett, W. J. Boone, A. L. Dagel, G. Dalton, W. B. Doriese, M. Durkin, J. W. Fowler, E. J. Garboczi, J. D. Gard, G. C. Hilton, J. Imrek, E. S. Jimenez, V. Y. Kotsubo, K. Larson, Z. H. Levine, J. A. B. Mates, D. McArthur, K. M. Morgan, N. Nakamura, G. C. O'Neil, N. J. Ortiz, C. G. Pappas, C. D. Reintsema, D. R. Schmidt, D. S. Swetz, K. R. Thompson, J. N. Ullom, C. Walker, J. C. Weber, A. L. Wessels, and J. W. Wheeler, "Design of a 3000 pixel transition-edge sensor X-ray spectrometer for microcircuit tomography," *IEEE Trans. Appl. Superconductivity*, vol. 31, pp. 1–5, 2021.

[111] R. Manekar, K. Tayal, V. Kumar, and J. Sun, "End-to-end learning for phase retrieval," in *ICML workshop on ML Interpretability for Scientific Discovery*, 2020.

[112] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pp. 1–6, IEEE, 2018.

[113] S. Marchesini and A. Sakdinawat, "Shaping coherent X-rays with binary optics," *Optics Express*, vol. 27, no. 2, pp. 907–917, 2019.

[114] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[115] I. Kang, "High-fidelity inversion at low-photon counts using deep learning and random phase modulation," Master's thesis, Massachusetts Institute of Technology, 2020.

[116] Z. H. Levine, A. J. Kearsley, and J. G. Hagedorn, "Bayesian tomography for projections with an arbitrary transmission function with an application in electron microscopy," *Journal of Research of the National Institute of Standards and Technology*, vol. 111, no. 6, pp. 411–417, 2006.

[117] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[118] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[119] M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, J. Bruna, V. Lostanlen, M. Chaudhary, M. J. Hirn, E. Oyallon, S. Zhang, C. Cella, and M. Eickenberg, "Kymatio: Scattering transforms in Python," *Journal of Machine Learning Research*, vol. 21, no. 60, pp. 1–6, 2020.

[120] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. Courville, "Modulating early visual processing by language," *arXiv:1707.00683*, 2017.

[121] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv:1610.07629*, 2016.

[122] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[123] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[124] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[125] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[126] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14985–15006, 2018.

[127] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," *arXiv:1704.00028*, 2017.

[128] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *arXiv:1809.11096*, 2018.

[129] Z. Lin, V. Sekar, and G. Fanti, "Why spectral normalization stabilizes GANs: Analysis and improvements," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[130] H. Lan, the Alzheimer Disease Neuroimaging Initiative, A. W. Toga, and F. Sepehrband, "Sc-gan: 3d self-attention conditional gan with spectral normalization for multi-modal neuroimaging synthesis," *bioRxiv*, 2020.

[131] I. Kavalerov, W. Czaja, and R. Chellappa, "A multi-class hinge loss for conditional GANs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1290–1299, 2021.