Learning Illumination Patterns for Coded Diffraction Phase Retrieval

Zikui Cai
ECE Department
UC Riverside
zcai032@ucr.edu

Rakib Hyder
ECE Department
UC Riverside
rhyde001@ucr.edu

M. Salman Asif
ECE Department
UC Riverside
sasif@ece.ucr.edu

Abstract

Signal recovery from nonlinear measurements involves solving an iterative optimization problem. In this paper, we present a framework to optimize the sensing parameters to improve the quality of the signal recovered by the given iterative method. In particular, we learn illumination patterns to recover signals from coded diffraction patterns using a fixed-cost alternating minimization-based phase retrieval method. Coded diffraction phase retrieval is a physically realistic system in which the signal is first modulated by a sequence of codes before the sensor records its Fourier amplitude. We represent the phase retrieval method as an unrolled network with a fixed number of layers and minimize the recovery error by optimizing over the measurement parameters. Since the number of iterations/layers are fixed, the recovery incurs a fixed cost. We present extensive simulation results on a variety of datasets under different conditions and a comparison with existing methods. Our results demonstrate that the proposed method provides near-perfect reconstruction using patterns learned with a small number of training images. Our proposed method provides significant improvements over existing methods both in terms of accuracy and speed.

1 Introduction

The problem of signal recovery from nonlinear measurements arises in various imaging and signal processing tasks [1–5]. Conventional methods for solving such inverse problems use an iterative method to recover the signal from given measurements. In this paper, we present a framework to optimize over the measurement parameters to improve the quality of signals recovered by the given iterative method. In particular, we learn illumination patterns to recover the signal from coded diffraction patterns (CDP) using a fixed-cost alternating minimization method.

Coded diffraction imaging is a specific instance of Fourier phase retrieval problems. Phase retrieval refers to a broad class of nonlinear inverse problems where we seek to recover a complex- (or real-) valued signal from its phase-less (or sign-less) measurements [1, 6–10]. In practice, these problems often arise in coherent optical imaging where an image sensor records the intensity of the Fourier measurements of the object of interest. In coded diffraction imaging, the signal of interest gets modulated by a sequence of known illumination patterns/masks before observing the Fourier intensity at the sensor [1, 7, 9]. Applications include X-ray crystallography [3, 11, 12], astronomy [13, 14], microscopy [4, 15–17], speech processing and acoustics [18–21], and quantum mechanics [22, 23].

We can model the sensor measurements for coded diffraction imaging as follows. Let us denote the signal of interest as $x \in \mathbb{R}^n$ or $\mathbb{C}^n$ that is modulated by T illumination patterns $D = \{d_1, \ldots, d_T\}$, where $d_t \in \mathbb{R}^n$ or $\mathbb{C}^n$. The amplitude of sensor measurements for tth illumination pattern can be written as

$$y_t = |\mathcal{F}(d_t \odot x)|, \tag{1}$$

Preprint. Under review.

where $\mathcal{F}$ denotes the Fourier transform operator and $\odot$ denotes an element-wise product. We note that real sensor measurements are proportional to the intensity of the incoming signal (i.e., square of the Fourier transform). In practice, however, solving the inverse problem with (non-square) amplitude measurements provides better results; therefore, we use the amplitude measurements throughout the paper.

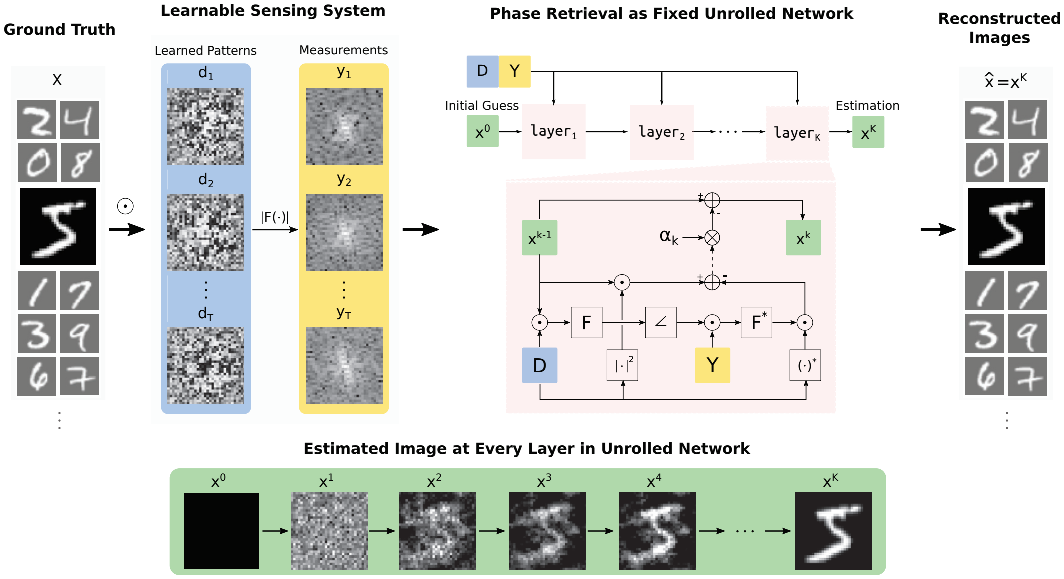


Figure 1: The block diagram of our proposed framework to learn illumination patterns while solving coded diffraction imaging using K iterations of a phase retrieval algorithm. The iterative algorithm is represented as an unrolled network with K layers. Steps at every iteration are fixed and depicted as an unrolled network (details can be found in Algorithm 2). Phase retrieval algorithm uses the measurements $Y = \{y_t\}$ and illumination patterns $D = \{d_t\}$ to provide an estimate x^K after K iterations. Our goal is to learn the illumination patterns D to minimize the error between the estimated x^K and the ground truth.

To recover the signal x from the nonlinear measurements, we can solve the following optimization problem:

$$\min_{x} \sum_{t=1}^{T} \|y_t - |\mathcal{F}(d_t \odot x)|\|_2^2. \tag{2}$$

In recent years, a number of iterative algorithms have been proposed for solving the problem in (2), which includes lifting-based convex methods, alternating minimization-based nonconvex methods, and greedy methods [7, 24–27].

Our goal is to learn a set of illumination patterns to optimize the recovery of an alternating minimization (AltMin) algorithm for solving the problem in (2). The AltMin method can be viewed as an unrolled gradient descent network, as shown in Fig. 1, where we fix the steps at every iteration and the total number of iterations for AltMin. One forward pass through the unrolled network is equivalent to K iterations of the AltMin algorithm. To minimize the computational complexity of the recovery algorithm, we keep the total number of iterations very small (e.g., $K = 50$). At the training stage, we optimize over the illumination patterns to minimize the error between the AltMin outputs after K iterations and the ground truth training images. At the test time, we solve the problem in (2) using K AltMin iteration with the learned illumination patterns (equivalent to one forward pass). We evaluated our method on different image datasets and compared against existing methods for coded diffraction imaging. We demonstrate that our proposed method of designing illumination patterns for a fixed-cost algorithm outperforms existing methods both in terms of accuracy and speed.

2 Related Work

Phase Retrieval and Coded Diffraction Patterns: Fourier phase retrieval problem arises in a number of imaging systems because standard image sensors can only record intensity of the observed measurements. This problem has been extensively studied over last five decades in optics, signal processing, and optimization [1–4, 8, 28–30]. Coded diffraction imaging is a physically realistic setup in which we can first modulate by signal of interest and then collect the intensity measurements [5, 31]. The modulation can be performed using a spatial light modulator or custom transparencies [4, 16, 32]. The recovery problems involves solving a phase retrieval problem; the presence of modulation patterns makes this a more tractable problem compared to classical Fourier phase retrieval [5].

The algorithms for solving phase retrieval problem can be broadly divided into non-convex and convex methods. Classical algorithms for phase retrieval rely on solving the underlying non-convex problem using alternating minimization [8, 25, 33]. Amplitude flow [34, 35], Wirtinger flow [36–39], alternating minimization (AltMin) [25–27, 37] are recent methods that solve the non-convex problem. Convex methods usually lift the nonconvex problem of signal recovery from quadratic measurements into a convex problem of low-rank matrix recovery from linear measurements [7, 40, 41]. The PhaseLift algorithm [7] and its variations [5, 24] can be considered under this class. We can also incorporate prior knowledge about the signal structure (e.g., sparsity, support, or positivity) in the recovery process constraints [25, 34, 39, 42–45].

Data-Driven Approaches for Phase Retrieval: A number of papers have recently explored the idea of replacing the classical (hand-designed) signal priors with deep generative priors for solving inverse problems [46–49]. Some of the generative prior-based approaches for phase retrieval are presented in [26, 29, 50–52].

Another growing trend is to apply deep learning to solve inverse problems (including phase retrieval) in an end-to-end manner, where deep networks are trained to learn a mapping from sensor measurements to the signal of interest using a large number of measurement-signal pairs. A few examples demonstrating the benefit of the data-driven approaches include robust phase retrieval [53], Fourier ptychographic microscopy [54], holographic image reconstruction [55], and correlography for non-line-of-sight imaging [56].

While our method is partially driven by data, our goal is not to learn a signal prior or a mapping from measurements to signal. We use data to learn the illumination patterns for a fixed recovery algorithm. The number of training images required by our method is extremely small (32 or 128 images only). Furthermore, the patterns we learn on one class of images provide good results on other types of images (see Table 2). Apart from the great flexibility, our method uses a well-defined AltMin routine, where we know exact steps for every iteration as opposed to the black-box deep models.

Unrolled Network for Inverse Problem: Iterative methods for solving the inverse problems, such as AltMin or other first-order methods, can be represented as unrolled networks. Every layer of such a network performs the same steps as a single iteration of the original method [57–66]. Some parameters of the iterative steps can be learned from data (e.g., step size, denoiser, or threshold parameters) but the basic structure and physical forward model are kept intact.

Learn to Sense: Deep learning methods have also been recently used to design the sensing system; especially in the context of compressive sensing and computational imaging [67–70]. The main objective in these methods is similar to ours, which is to select sensor parameters to recover best possible signal/image from the sensor measurements. The sensor parameters may involve selection of samples/frames, design of sampling waveforms, or illumination patters as we discuss in this paper. In principle, the sensor can be treated as the first layer of the network with some physical constraints on the parameters [57]. In contrast to most of the existing methods that learn a deep network to solve the inverse problem, our method uses a predefined iterative method as an unrolled network while learning the illumination patterns using a small number of training images.

3 Proposed Method

We use N training images $(x_1, \dots, x_N)$ to learn T illumination patterns that provide best reconstruction using a predefined (iterative) phase retrieval algorithm. Furthermore, to ensure that the illumination patterns are physically realizable, we constrain their values to be in the range $[0, 1]$. We use a sigmoid function over unconstrained parameters $\Theta = \{\theta_1, \dots, \theta_T\}$ to define the illumination patterns; that is, $d_t = \text{sigmoid}(\theta_t)$ for all $t = 1, \dots, T$.

Our proposed method for learning illumination patterns can be divided into two parts: The first (inner) part involves solving the phase retrieval problem with given coded diffraction patterns using

Algorithm 1 Learning illumination patterns

```
Input: Training set $X$ with $N$ images $X = \{x_1, \dots, x_N\}$.
Initialize: Initialize the optimization variables for $T$ patterns as $\Theta = \{\theta_1, \dots, \theta_T\}$ from a uniform distribution.
for epoch $= 1, 2, ..., M$ do                                              ▷ M epochs
    Generate illumination patterns in the range [0,1] as $d_t = \text{sigmoid}(\theta_t)$ for all $t$.
    for $n = 1, 2, ..., N$ do                                              ▷ N samples
        $Y^n = \{y_1^n, \dots, y_T^n \mid y_t^n = |\mathcal{F}(d_t \odot x_n)|\}$
        $x_n^K(\Theta) \leftarrow$ solveCDP($Y^n$,$D$)                      ▷ solveCDP for each sample
    end for
    $L_\Theta = \sum_{n=1}^N \|x_n - x_n^K(\Theta)\|_2^2$
    $\Theta \leftarrow \Theta - \beta \nabla_\Theta L_\Theta$                ▷ Optimize over $\Theta$
end for
Output: Optimal illumination patterns $D = \{d_1, \dots, d_T \mid d_t = \text{sigmoid}(\theta_t)\}$.
```

Algorithm 2 `solveCDP(`Y, D`)` via AltMin

```
Input: Measurements $Y = \{y_1, \dots, y_t\}$ and illumination patterns $D = \{d_1, \dots, d_T\}$.
Initialization: Initialize estimate $x^0 = 0$.
for $k = 1, 2, ..., K$ do                                                  ▷ K iterations of AltMin
    $p_t^{k-1} \leftarrow \text{sign}(\mathcal{F}(d_t \odot x^{k-1})$
    $\nabla_x L_{x,p} = \frac{2}{T} \sum_{t=1}^T [|d_t|^2 \odot x^{k-1} - d_t^* \odot \mathcal{F}^*(p_t^{k-1} \odot y_t)]$
    $x^k \leftarrow x^{k-1} - \alpha \nabla_x L_{x,p}$
end for
Output: Estimated signal $x^K$
```

AltMin as an unrolled network (see block diagram in Fig. 1); Second part is updating the illumination patterns based on backpropagating the image reconstruction loss. These two parts together yield optimal image reconstruction and illumination patterns. Pseudocodes for both parts are listed in Algorithms 1,2.

Phase retrieval as alternating minimization (AltMin): Given measurements $Y = \{y_1, \dots, y_T\}$ and illumination patterns $D = \{d_1, \dots, d_T\}$, we seek to solve the CDP phase retrieval problem by minimizing the loss function defined in (2) as

$$L_x = \frac{1}{T} \sum_{t=1}^{T} \|y_t - |\mathcal{F}(d_t \odot x)|\|_2^2. \tag{3}$$

Even though the loss function in (3) is nonconvex and nonsmooth with respect to x, we can minimize it using the well-known alternating minimization (AltMin) with a single gradient descent step [25,37]. We define a new variable for the estimated phase of linear measurements as $p_t = \text{phase}(\mathcal{F}(d_t \odot x))$ and reformulate the loss function in (3) into

$$L_{x,p} = \frac{1}{T} \sum_{t=1}^{T} \|p_t \odot y_t - \mathcal{F}(d_t \odot x)\|_2^2. \tag{4}$$

The gradient step can be computed as

$$\nabla_x L_{x,p} = \frac{2}{T} \sum_{t=1}^{T} |d_t|^2 \odot x - d_t^* \odot \mathcal{F}^*(p_t \odot y_t). \tag{5}$$

We initialize the image estimate as a zero vector and update the estimate at every iteration as $x^k = x^{k-1} - \alpha_{k-1} \nabla_x L_{x,p}$, where α_{k-1} denotes the step size. In our implementation, we used a fixed step size α for all iterations. We use K iterations of the gradient descent in the unrolled network and denote the final estimate as x^K. We summarize the steps of CDP phase retrieval routine as `solveCDP` in Algorithm 2.

Learning illumination patterns: To learn a set of illumination patterns that provide the best reconstruction with the predefined iterative method (or the unrolled network), we seek to minimize the difference between the original training images and their estimates. In this regard, we minimize the following quadratic loss function with respect to Θ:

$$L_\Theta = \sum_{n=1}^{N} \|x_n - x_n^K(\Theta)\|_2^2, \tag{6}$$

where $x_n^K(\Theta)$ denotes the `solveCDP` estimate of nth training image for the given values of Θ. Note that for a given values of Θ, we can define illumination patterns as $d_t = \text{sigmoid}(\theta_t)$ and sensor measurements for x_n as $y_t^n = |\mathcal{F}(d_t \odot x_n)|$ for $t = 1, \dots, T$ and $n = 1, \dots, N$. We use Adam optimizer in PyTorch [71, 72] to minimize the loss function in (6). A summary of the algorithm for learning the illumination patterns is also listed in Algorithm 1.

4 Experiments

Datasets. We used MNIST digits, Fashion MNIST (F. MNIST), CIFAR10, SVHN and CelebA datasets for training and testing in our experiments. We use 128 images from each of the datasets for training and another 1000 images for testing. To make the tiny-image datasets uniform, we reshaped all of them to 32 × 32 size with grayscale values. Images in CelebA dataset have 218 × 178 pixels, we first converted all the images to grayscale, cropped 178 × 178 region in the center, and resized to 200 × 200. In Fig. 5, we evaluated the performance of our method on images used in [53].

Measurements. We use the amplitude of the 2D Fourier transform of the images modulated with T illumination patterns as the measurements. Unless otherwise mentioned, we used noiseless measurements. We report results for measurements with Gaussian and Poisson noise in Fig. 6.

Computing Infrastructure. Intel Core i7-8700 CPU and NVIDIA TITAN Xp GPU.

4.1 Setup and hyper-parameter search

The hyper-parameters include the number of iterations (K), step size α, and the number of training samples N. We set the default value of $K = 50$, but we will show later that K can be adjusted as a trade-off between better reconstruction quality and shorter run time. We tested all methods for $T = \{2, 4, 8\}$ to evaluate three cases where signal recovery is hard, moderate, and easy, respectively. Through grid search, we found that it provides the best results over all datasets when $\alpha * T = 4$. We also studied the effect of the number of training images and found that illumination patterns learned on 32 randomly selected images provide good recovery over the entire dataset. The test accuracy improves slightly as we increase the number of training samples. To be safe, we used 128 training images in all our experiments.

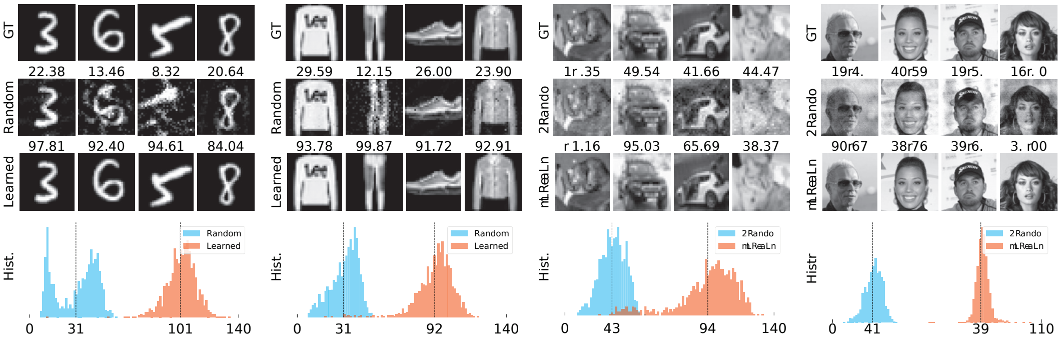
4.2 Comparison between random and learned illumination patterns

To demonstrate the advantages of our learned illumination patterns, we compare the performance of learned and random illumination patterns on five different datasets. We learn a set of $T = \{2, 4, 8\}$ illumination patterns on 128 training images from a dataset and test them on 1000 test images from the same dataset. For random patterns, we draw T independent patterns from Uniform(0,1) distribution and test their performance on the same 1000 samples that we used for the learned case. We repeat this process 30 times and choose the best result to compare with the results for the learned illumination patterns. The average PSNR over all 1000 test image reconstructions is presented in Table 1, which shows that the learned illumination patterns perform significantly better than the random patterns for all values of T. In addition to that, we can observe a transition in the performance for $T = 4$, where random patterns provide poor quality reconstructions and learned patterns provide very high quality reconstructions.

To highlight this effect, we show a small set of reconstructed images and histograms of PSNRs of all the reconstructed images from learned and random illumination patterns in Fig. 2 for $T = 4$ patterns. The result suggests that the learned illumination patterns demonstrate consistently better performance compared to random illumination patterns.

Table 1: Average PSNR for learned and random illumination patterns tested on different datasets.

PSNR	2 Illumination Patterns		4 Illumination Patterns		8 Illumination Patterns	
	Random	Learned	Random	Learned	Random	Learned
MNIST	13.59	27.77	28.61	101.50	46.43	111.79
F. MNIST	16.26	25.17	30.05	92.44	52.05	108.85
CIFAR10	13.40	22.93	27.51	82.02	60.64	104.61
SVHN	12.24	20.27	26.24	84.43	65.33	109.25
CelebA	12.67	18.49	21.67	78.98	41.49	90.29

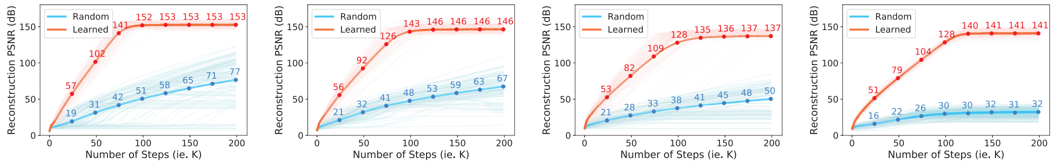


(a) MNIST **(b)** F. MNIST **(c)** CIFAR10 **(d)** CelebA

Figure 2: Selected ground truth (GT) images, corresponding reconstructed images using random and learned illumination patterns. PSNR is shown on top of every reconstruction. Below each dataset, we show the histograms of the PSNRs of all images with random patterns (shown in blue) and learned patterns (shown in orange). The dashed vertical line indicates the mean of all PSNRs. We used $T = 4$ illumination patterns. Random illumination patterns are selected best out of 30 trials. The learned illumination patterns are trained on 128 training images.

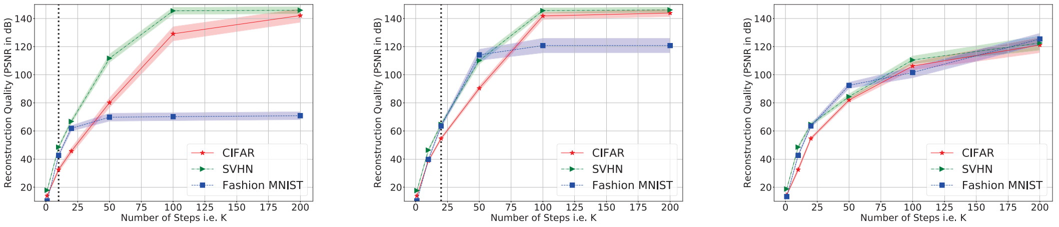
4.3 Effect of number of iterations/layers (K)

Figure 3 shows the performance of the learned and random illumination patterns as we increase K to 200 at test time using the patterns learned for $K = 50$. We observed that, with the learned patterns, the image reconstruction process converges faster and is more stable (smaller variance) than the case with random patterns. The red curve in Fig. 3 has a steeper slope and narrower shades. Besides the default setting for $K = 50$, we also learn the illumination patterns for different values of K. Figure 4 shows that we can recover images in a small number of iterations if we use learned illumination patterns. We also observe that we can perform better if we use more iterations in testing than in training. We have chosen $K = 50$ for most of the experiments as a trade-off between computational cost and reconstruction performance.



(a) MNIST **(b)** F. MNIST **(c)** CIFAR10 **(d)** CelebA

Figure 3: Comparison of the reconstruction quality with random and learned illumination patterns for different values of $K = 1, \ldots, 200$. We plot the average PSNR in bright color and the PSNR of randomly selected 100 samples in light shadows. **Learned** represents the reconstruction PSNR with learned illumination patterns (shown in red), and **Random** represents PSNR for random illumination patterns (shown in blue). The number of illumination patterns is $T = 4$. Random illumination patterns are selected best out of 30 trials. The learned illumination patterns are trained on 128 training images and number of iterations $K = 50$ during training.



(a) Training K=10 **(b)** Training K=20 **(c)** Training K=Test K

Figure 4: Reconstruction quality vs number of iterations (layers) at test time (i.e., K is different for training and testing with $T = 4$). We show error bar of $\pm 0.25\sigma$ for each dataset. In (a) and (b), we fixed K (K=10, 20) and tested using different K. In (c), we trained and tested using the same number of layers.

Table 2: Reconstruction performance of illumination patterns learned and tested on different datasets. Every column corresponds to patterns learned on a fixed dataset and tested on all. Rand column reports the performance of random illumination patterns.

Test	4 Illumination Patterns					8 Illumination Patterns				
\Train	MNIST	FMNIST	CIFAR	SVHN	Rand	MNIST	FMNIST	CIFAR	SVHN	Rand
MNIST	101.50	83.13	78.04	54.83	28.61	111.79	92.66	82.75	74.86	46.43
FMNIST	64.47	92.44	85.14	76.41	30.05	82.37	108.85	85.80	93.09	52.05
CIFAR	31.77	46.79	82.02	68.12	27.51	52.75	84.94	104.61	87.85	60.64
SVHN	40.04	55.31	87.77	84.43	26.24	65.88	100.00	104.69	109.25	65.33

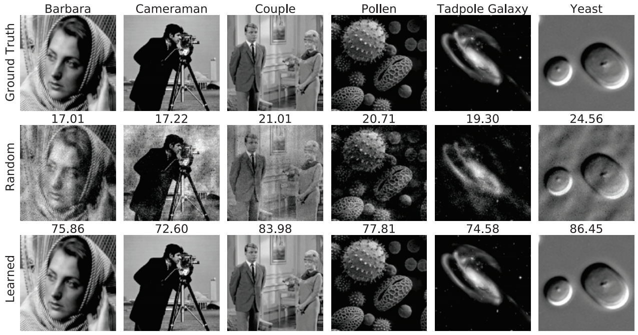


Figure 5: First Row: Ground truth images from image processing standard test datasets. **Second Row:** Reconstruction using random illumination patterns with uniform random distribution [0, 1] (we selected $T = 4$ patterns that provided best results on celebA test images in **30 trials**). PSNR numbers are shown on the top of reconstructed images. **Third Row:** Reconstruction using the patterns trained on celebA dataset. Each image has 200×200 pixels and the number of illumination patterns is $T = 4$.

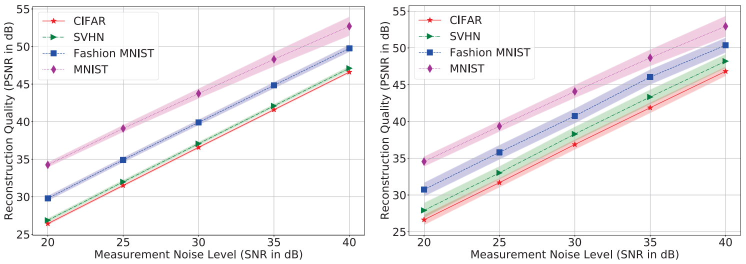
4.4 Generalization of learned patterns on different datasets

To explore the generalizability of our learned illumination patterns, we use patterns learned on one dataset to recover images from another. The results are shown in Table. 2. As we can see in the table, the diagonal numbers are generally the best, and off-diagonal numbers are generally better than the ones with random illumination patterns.

We have also tested the learned illumination patterns on several classical images. Some results are shown in Fig. 5. We used illumination patterns learned on 128 celebA images, but we can see that the learned illumination patterns perform better than the randomly chosen illumination patterns for classical images which indicates some generalizability of our learned illumination patterns.

4.5 Robustness to noise

To investigate the robustness of our method to noise, we train our illumination patterns on noiseless measurements obtained from the training datasets. We then added Gaussian and Poisson noise at different levels to the measurements from the test datasets. Poisson noise or shot noise is the most



(a) Gaussian **(b)** Poisson

Figure 6: Reconstruction quality of the test images vs noise level of the measurements for different datasets. Here we show shaded error bar of $\pm 0.25\sigma$ for each dataset. We learn the illumination patterns ($T = 4$) on 128 noiseless training images of corresponding datasets.

Table 3: Reconstruction PSNR and run time (sec) of our and existing methods on different datasets.

Algorithm	MNIST		F. MNIST		CIFAR10		SVHN		CelebA	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
Fienup	13.17	0.31	23.28	0.30	32.79	0.29	40.57	0.29	31.57	5.42
GS	12.96	0.31	23.28	0.30	32.71	0.29	40.99	0.29	31.69	5.22
WirtFlow	10.74	0.07	13.72	0.06	16.28	0.05	19.65	0.04	16.93	1.26
AmpFlow	17.09	0.07	32.83	0.05	42.47	0.05	47.43	0.04	37.76	1.24
Kaczmarz	11.50	0.05	13.46	0.04	15.31	0.03	18.39	0.03	16.63	0.94
Deep Models	31.73	8.41	22.33	8.45	25.63	8.40	27.81	8.25	22.31	10.55
Ours	**101.50**	**0.02**	**92.44**	**0.02**	**82.02**	**0.02**	**84.43**	**0.02**	**78.98**	**0.04**

common in the imaging systems. We model the Poisson noise following the approach in [53]. Let us denote the i^{th} element of measurement vector corresponding to t^{th} illumination pattern, y_t as

$$y_t(i) = |z_t(i)| + \eta_t(i) \quad \forall i = 1, 2, \ldots, m \tag{7}$$

where $\eta_t(i) \sim \mathcal{N}(0, \lambda|z_t(i)|)$ and $z_t = \mathcal{F}(d_t \odot x)$. We varied λ to generate noise at different signal-to-noise ratios. Poisson noise affects larger values in measurements with higher strength than the smaller values. Since the sensors can measure only positive measurements, we kept the measurements positive by applying ReLU function after noise addition. We expect the reconstruction to be affected by noise as we did not use any denoiser. We observe the effect of noise in Figure 6. Even though noise affects the reconstructions, we can get reasonable reconstruction up to a certain level of noise. The relationship between noise level and reconstruction performance also indicates that our phase retrieval system is quite stable.

4.6 Comparision with existing methods

We compare our method with various existing methods on different datasets and show the reconstruction PSNR (dB) and run time (seconds) per image in Table. 3. The description of the datasets can be found in section 4. We compare with Fineup [6], Gerchberg-Saxton [28], Wirtinger Flow [36], Amplitude Flow [38], Kaczmarz [73] and deep generative models [52]. For [6, 28, 36, 38, 73], we use the code collected in PhasePack [31]. In PhasePack, we restricted all the illumination patterns in the range of $[0, 1]$ and set the maximum number of iterations to be 50 which is the same as our default setting. We note that all of the methods in PhasePack use optimal spectral initialization; in contrast, we only use zero initialization. For deep generative models, we use a modified version of the publicly available code for [52]. The code only provided pretrained DCGAN models for MNIST and F. MNIST; therefore, we trained our DCGAN models on the other datasets. This method is noticeably time-consuming because it optimizes over the latent vector for the deep model and uses 2000 iterations for each image where each iteration requires a forward and backward pass through the deep model. For all of the methods, we selected the best PSNR from 5 trials and report the average time. For our method, we report the run time with batch size equal to 1, which aligns with other methods. We observe from the table that our method with learned patterns performs significantly better than all the other algorithms in terms of both reconstruction quality and run time.

References

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

[2] A. Maiden and J. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," *Ultramicroscopy*, vol. 109, no. 10, pp. 1256–1262, 2009.

[3] R. Millane, "Phase retrieval in crystallography and optics," *JOSA A*, vol. 7, no. 3, pp. 394–411, 1990.

[4] J. M. Rodenburg, "Ptychography and related diffractive imaging methods," *Advances in imaging and electron physics*, vol. 150, pp. 87–184, 2008.

[5] E. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval from coded diffraction patterns," *Appl. Comput. Harmon. Anal.*, vol. 39, no. 2, pp. 277–299, 2015.

[6] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Optics letters*, vol. 3, no. 1, pp. 27–29, 1978.

[7] E. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Comm. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.

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

[9] K. Jaganathan, Y. C. Eldar, and B. Hassibi, "Phase retrieval: An overview of recent developments," *arXiv preprint arXiv:1510.07713*, 2015.

[10] B. Muminov and L. T. Vuong, "Small-brain neural networks rapidly solve inverse problems with vortex fourier encoders," *arXiv preprint arXiv:2005.07682*, 2020.

[11] J. D. Watson, F. Crick *et al.*, "A structure for deoxyribose nucleic acid," 1953.

[12] R. Harrison, "Phase problem in crystallography," *JOSA a*, vol. 10, no. 5, pp. 1046–1055, 1993.

[13] C. Fienup and J. Dainty, "Phase retrieval and image reconstruction for astronomy," *Image recovery: theory and application*, vol. 231, p. 275, 1987.

[14] R. A. Gonsalves, "Perspectives on phase retrieval and phase diversity in astronomy," in *Adaptive Optics Systems IV*, vol. 9148. International Society for Optics and Photonics, 2014, p. 91482P.

[15] D. Misell, "A method for the solution of the phase problem in electron microscopy," *Journal of Physics D: Applied Physics*, vol. 6, no. 1, p. L6, 1973.

[16] J. Miao, T. Ishikawa, Q. Shen, and T. Earnest, "Extending x-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes," *Annu. Rev. Phys. Chem.*, vol. 59, pp. 387–410, 2008.

[17] L. Tian, X. Li, K. Ramchandran, and L. Waller, "Multiplexed coded illumination for fourier ptychography with an led array microscope," *Biomedical optics express*, vol. 5, no. 7, pp. 2376–2389, 2014.

[18] L. Rabiner, "Fundamentals of speech recognition," *Fundamentals of speech recognition*, 1993.

[19] R. Balan, P. Casazza, and D. Edidin, "On signal reconstruction without phase," *Applied and Computational Harmonic Analysis*, vol. 20, no. 3, pp. 345–356, 2006.

[20] R. Balan, "On signal reconstruction from its spectrogram," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2010, pp. 1–4.

[21] K. Jaganathan, Y. C. Eldar, and B. Hassibi, "Stft phase retrieval: Uniqueness guarantees and recovery algorithms," *IEEE Journal of selected topics in signal processing*, vol. 10, no. 4, pp. 770–781, 2016.

[22] J. V. Corbett, "The pauli problem, state reconstruction and quantum-real numbers," *Reports on Mathematical Physics*, vol. 1, no. 57, pp. 53–68, 2006.

[23] H. Reichenbach, *Philosophic foundations of quantum mechanics*. Courier Corporation, 1998.

[24] D. Gross, F. Krahmer, and R. Kueng, "Improved recovery guarantees for phase retrieval from coded diffraction patterns," *Appl. Comput. Harmon. Anal.*, vol. 42, no. 1, pp. 37–64, 2017.

[25] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2013, pp. 2796–2804.

[26] R. Hyder, V. S., C. Hegde, and M. Asif, "Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7705–7709.

[27] G. Jagatap and C. Hegde, "Fast, sample-efficient algorithms for structured phase retrieval," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2017.

[28] R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.

[29] P. Hand, O. Leong, and V. Voroninski, "Phase retrieval under a generative prior," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2018, pp. 9154–9164.

[30] S. Nayer, P. Narayanamurthy, and N. Vaswani, "Phaseless pca: Low-rank matrix recovery from column-wise phaseless measurements," in *International Conference on Machine Learning*, 2019, pp. 4762–4770.

[31] R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, and T. Goldstein, "Phasepack: A phase retrieval library," *Asilomar Conference on Signals, Systems, and Computers*, 2017.

[32] G. Jagatap, Z. Chen, S. Nayer, C. Hegde, and N. Vaswani, "Sample efficient fourier ptychography for structured data," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 344–357, 2020.

[33] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and fienup variants: a view from convex optimization," *JOSA A*, vol. 19, no. 7, pp. 1334–1345, 2002.

[34] G. Wang, L. Zhang, G. B. Giannakis, M. Akcakaya, and J. Chen, "Sparse phase retrieval via truncated amplitude flow," *IEEE Trans. Signal Processing*, vol. 66, pp. 479–491, 2018.

[35] G. Wang and G. Giannakis, "Solving random systems of quadratic equations via truncated generalized gradient flow," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 568–576.

[36] E. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: theory and algorithms," *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[37] H. Zhang and Y. Liang, "Reshaped wirtinger flow for solving quadratic system of equations," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 2622–2630.

[38] Y. Chen and E. Candes, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2015, pp. 739–747.

[39] T. Cai, X. Li, Z. Ma *et al.*, "Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow," *Ann. Stat.*, vol. 44, no. 5, pp. 2221–2251, 2016.

[40] M. Soltani and C. Hegde, "Fast algorithms for demixing sparse signals from nonlinear observations," *arXiv preprint arXiv:1608.01234*, 2016.

[41] M. Fazel, E. Candes, B. Recht, and P. Parrilo, "Compressed sensing and robust recovery of low rank matrices," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE, 2008, pp. 1043–1047.

[42] H. Ohlsson, A. Yang, R. Dong, and S. Sastry, "Cprl–an extension of compressive sensing to the phase retrieval problem," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2012, pp. 1367–1375.

[43] X. Li and V. Voroninski, "Sparse signal recovery from quadratic measurements via convex programming," *SIAM J. on Math. Analysis*, vol. 45, no. 5, pp. 3019–3033, 2013.

[44] S. Bahmani and J. Romberg, "Efficient compressive phase retrieval with constrained sensing vectors," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2015, pp. 523–531.

[45] K. Jaganathan, S. Oymak, and B. Hassibi, "Recovery of sparse 1-d signals from the magnitudes of their fourier transform," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*. IEEE, 2012, pp. 1473–1477.

[46] A. Bora, A. Jalal, E. Price, and A. Dimakis, "Compressed sensing using generative models," *Proc. Int. Conf. Machine Learning*, 2017.

[47] P. Hand and V. Voroninski, "Compressed sensing from phaseless gaussian measurements via linear programming in the natural parameter space," *arXiv preprint arXiv:1611.05985*, 2016.

[48] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[49] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, “Compressed sensing with deep image prior and learned regularization,” *arXiv preprint arXiv:1806.06438*, 2018.

[50] G. Jagatap and C. Hegde, “Algorithmic guarantees for inverse imaging with untrained network priors,” *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2019.

[51] F. Shamshad and A. Ahmed, “Robust compressive phase retrieval via deep generative priors,” *arXiv preprint arXiv:1808.05854*, 2018.

[52] C. A. Metzler and G. Wetzstein, “Deep s^{3}pr: Simultaneous source separation and phase retrieval using deep generative models,” *arXiv preprint arXiv:2002.05856*, 2020.

[53] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, “prdeep: Robust phase retrieval with a flexible deep network,” *arXiv preprint arXiv:1803.00212*, 2018.

[54] M. Kellman, E. Bostan, M. Chen, and L. Waller, “Data-driven design for fourier ptychographic microscopy,” *International Conference for Computational Photography*, 2019.

[55] 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, no. 2, pp. 17 141–17 141, 2018.

[56] C. A. Metzler, F. Heide, P. Rangarajan, M. M. Balaji, A. Viswanath, A. Veeraraghavan, and R. G. Baraniuk, “Deep-inverse correlography: towards real-time high-resolution non-line-of-sight imaging,” *Optica*, vol. 7, no. 1, pp. 63–71, 2020.

[57] M. R. Kellman, E. Bostan, N. A. Repina, and L. Waller, “Physics-based learned design: optimized coded-illumination for quantitative phase imaging,” *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 344–353, 2019.

[58] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, “Unrolled optimization with deep priors,” *arXiv preprint arXiv:1705.08041*, 2017.

[59] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proc. Int. Conf. Machine Learning*, 2010, pp. 399–406.

[60] S. Wang, S. Fidler, and R. Urtasun, “Proximal deep structured models,” in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 865–873.

[61] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a variational network for reconstruction of accelerated mri data,” *Magnetic resonance in medicine*, vol. 79, no. 6, pp. 3055–3071, 2018.

[62] J. Sun, H. Li, Z. Xu *et al.*, “Deep admm-net for compressive sensing mri,” in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 10–18.

[63] U. S. Kamilov and H. Mansour, “Learning optimal nonlinearities for iterative thresholding algorithms,” *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 747–751, 2016.

[64] E. Bostan, U. S. Kamilov, and L. Waller, “Learning-based image reconstruction via parallel proximal algorithm,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 989–993, 2018.

[65] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *arXiv preprint arXiv:1912.10557*, 2019.

[66] D. Liang, J. Cheng, Z. Ke, and L. Ying, “Deep mri reconstruction: Unrolled optimization algorithms meet neural networks,” *arXiv preprint arXiv:1907.11711*, 2019.

[67] A. Mousavi and R. G. Baraniuk, “Learning to invert: Signal recovery via deep convolutional networks,” *arXiv preprint arXiv:1701.03891*, 2017.

[68] S. Wu, A. Dimakis, S. Sanghavi, F. Yu, D. Holtmann-Rice, D. Storcheus, A. Rostamizadeh, and S. Kumar, “Learning a compressed sensing measurement matrix via gradient unrolling,” in *Proc. Int. Conf. Machine Learning*, 2019, pp. 6828–6839.

[69] A. W. Bergman, D. B. Lindell, and G. Wetzstein, “Deep Adaptive LiDAR: End-to-end Optimization of Sampling and Depth Completion at Low Sampling Rates,” *Proc. IEEE ICCP*, 2020.

[70] J. Wang, Q. Gao, X. Ma, Y. Zhao, and Y. Fang, “Learning to sense: Deep learning for wireless sensing with less training efforts,” *IEEE Wireless Communications*, 2020.

[71] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*.

[73] K. Wei, “Solving systems of phaseless equations via kaczmarz methods: A proof of concept study,” *Inverse Problems*, vol. 31, no. 12, p. 125008, 2015.