DATA-DRIVEN ILLUMINATION PATTERNS FOR CODED DIFFRACTION IMAGING

Zikui Cai, Rakib Hyder, and M. Salman Asif

Department of Electrical and Computer Engineering
University of California Riverside

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 runs under 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.

Index Terms— Coded diffraction imaging, fourier phase retrieval, optimization, illumination pattern.

1. INTRODUCTION

The problem of signal recovery from nonlinear measurements arises in various imaging and signal processing tasks [1–3]. Conventional methods for solving such inverse problems use iterative methods to recover the signal from given measurements. In this paper, we present a framework to optimize 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.

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 $D = \{d_1, \dots, d_T\}$, where $d_t \in \mathbb{R}^n$ or $\mathbb{C}^n$. The amplitude of sensor measurements for t^{th} illumination pattern can be written as

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

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 [4, 5]; therefore, we use the amplitude measurements throughout this paper.

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 [6–9].

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, 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. We can increase or decrease the number of iterations for better accuracy or faster run-time. To minimize the computational complexity of the recovery algorithm, we keep the total number of iterations 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.

The key contributions of this paper are summarized as follows.

- We learned illumination patterns for a non-linear inverse problem (coded diffraction imaging) using unrolled network formulation of a classical AltMin method.
- We showed that with our designed patterns and unrolled AltMin method outperform computationally complex algorithms and provide superior image reconstruction.
- Our algorithm requires only a small number of training images to learn the illumination patterns. It is crucial in applications because finding training samples is difficult in practice.
- Our learned illumination patterns can also help other algorithms achieve higher performance even though they are not used for training.

2. RELATED WORK

Phase Retrieval and Coded Diffraction Patterns: Coded diffraction imaging is a physically realistic setup of Fourier phase retrieval problem, where we can first modulate signal of interest and then collect the intensity measurements [3, 10]. The presence of modulation

patterns makes this a more tractable problem compared to classical Fourier phase retrieval [3]. The algorithms for solving phase retrieval problem can be broadly divided into non-convex and convex methods. Amplitude flow [11], Wirtinger flow [12], alternating minimization (AltMin) [8] 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 [6, 13]. The PhaseLift algorithm [6] and its variations [3, 7] can be considered under this class. Other algorithms, such as PhaseMax [14] and PhaseLin [15], use convex relaxation to solve non-convex phase retrieval problem without lifting the problem to a higher dimension.

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 [16–18]. 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.

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 (128 images only). Furthermore, the patterns we learn on one class of images provide good results on other types of images. 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 [19,20]. 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. In our recent work [21], we used the idea of unrolled network to solve phase retrieval problem from the holographic measurements.

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 [22, 23]. The main objective, similar to ours, is to select sensor parameters to recover best possible signal/image from the sensor measurements. This may involve selection of samples/frames or illumination patters as we discuss in this paper. 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. In principle, the sensor can be treated as the first layer of the network with some physical constraints on the parameters [24]. The method in [24] uses unrolled network to learn the sensing parameters for quantitative phase imaging problem under the "weak object approximation". This approximation turns the original nonlinear problem into a linear inverse problem. In our setup, we do not make any such assumptions on target object and solve the nonlinear inverse problem.

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 $t = 1, \dots, T$.

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 $d_t = \text{sigmoid}(\theta_t)$ 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`)`
  end for
  $L_\Theta = \sum_{n=1}^{N} \|x_n - x_n^K(\Theta)\|_2^2$
  $\Theta \leftarrow \Theta - \beta \nabla_\Theta L_\Theta$ ▷ Update Θ with stepsize β
end for
Output: Learned illumination patterns

$$D = \{d_1, \dots, d_T \mid d_t = \text{sigmoid}(\theta_t)\}.$$

Algorithm 2 `solveCDP(`Y, D`)` via alternating minimization using single-step gradient descent

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

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 AltMin as an unrolled network; Second part is updating the illumination patterns based on backpropagating the image reconstruction loss. These two parts provide optimized 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 gradient descent [8, 25]. 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 with respect to x can be computed as

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

where $\mathcal{F}^*$ denotes the inverse Fourier transform and d_t^* is the conjugate of pattern d_t. We can update the estimate at every iteration as

$$x^k = x^{k-1} - \alpha_{k-1}\nabla_x L_{x,p}, \quad (6)$$

where α_{k-1} denotes the step size. Another way is to directly solve for x^k such that $\nabla_x L_{x,p} = 0$. The closed-form solution is

$$x^k = (\sum_{t=1}^{T} |d_t|^2)^{-1} \odot [\sum_{t=1}^{T} d_t^* \odot \mathcal{F}^*(p_t^{k-1} \odot y_t)]. \quad (7)$$

We compared these 2 strategies and found that single-step gradient descent tends to work well in practice and the closed-form solution does not show advantage over the single-step gradient descent. In our implementation, we used the former strategy (Algorithm 2) and fixed a step size α for all iterations. The unrolled network has K layers that implement K iterations of the gradient descent, and the final estimate is denoted as x^K.

Choice of initialization is important, and our method can handle different types of initialization. Zero initialization, where every pixel of the initial guess of x^0 is 0, is the simplest and cost-free method. Many recent phase retrieval algorithms [12, 25–27] use spectral initialization, which tries to find a good initial estimate. However, it requires computing the principal eigenvector of the following positive semidefinite matrix, $\sum_{t=1}^{T} \text{diag}(d_t^*)\mathcal{F}^*\text{diag}(|y_t|^2)\mathcal{F}\text{diag}(d_t)$. In our experiments, we observed that spectral initialization does not provide a significant improvement in terms of image reconstruction, and that our algorithm can perform very well using the overhead-free zero initialization.

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:

$$L_\Theta = \sum_{n=1}^{N} \|x_n - x_n^K(\Theta)\|_2^2, \quad (8)$$

where $x_n^K(\Theta)$ denotes the `solveCDP` estimate of nth training image for the given values of Θ. Note that for 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 $t = 1, \ldots, T$ and $n = 1, \ldots, N$. We use Adam optimizer in PyTorch [28, 29] to minimize the loss function in (8). A summary of the algorithm for learning the illumination patterns is also listed in Algorithm 1.

4. EXPERIMENTS

Datasets. We used MNIST digits and CelebA datasets for training and testing in our experiments. We used 128 images from each of the datasets for training and another 1000 images for testing. 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. Furthermore, we report the performance of our method on images used in [5] in Fig. 2.

Measurements. We used 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.

Computing platform. We performed all the experiments using a computer equipped with 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 K can be adjusted as a trade-off between better reconstruction quality and shorter run time. We tested all methods for $T = \{2, 3, 4, 8\}$ to evaluate cases where signal recovery is hard, moderate, and easy. Through grid search, we found that it provides the best results over all datasets when $\alpha = 4/T$.

4.2. Comparison between random and learned 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 $T = \{2, 3, 4, 8\}$ 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 $T = 3$, where random patterns provide poor quality reconstructions and learned patterns provide reasonably high quality reconstructions. Furthermore, the learned patterns provide very high quality reconstructions for $T \geq 4$.

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. 1 for $T = 4$ patterns. The result suggests that the learned illumination patterns demonstrate consistently better performance compared to random illumination patterns.

4.3. Comparision with existing methods

We compare our method with various existing methods using different datasets. These existing methods fall into 4 categories:

- Hybrid input output (HIO) [31] and Gerchberg-Saxton (GS) [32] (alternating minimization methods)
- Wirtinger Flow [12] and Amplitude Flow [26] (non-convex, gradient descent-based methods)
- PhaseMax [27] (a convex method)
- Deep S^{3}PR [30] (deep model-based method).

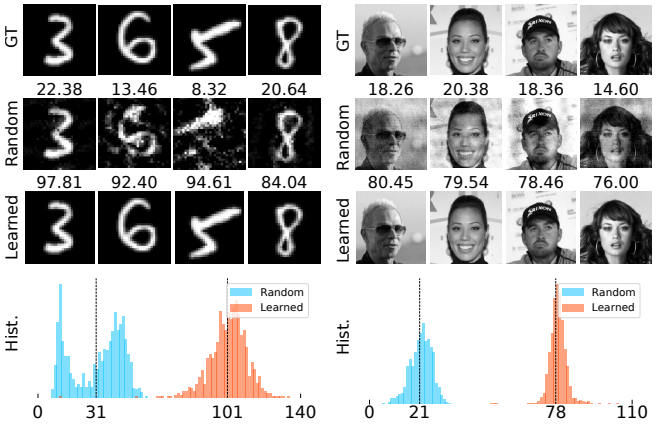
We compare the performance of our method with these methods in terms of reconstruction quality.

For algorithms in [12, 26, 27, 31, 32], we used PhasePack [10] package with default spectral initialization. In our comparison, we used $T = 4$ illumination patterns and in value range of $[0, 1]$. To make a fair comparison between our models in reconstruction quality, we set the error tolerance ($\texttt{tol} = 10^{-6}$) and run each algorithm till convergence.

For deep generative models, we used a modified version of the publicly available code and DCGAN model for MNIST from [30] and

Table 1: PSNR (`mean` $\pm$ `std`) for random and learned illumination patterns tested on different datasets.

Dataset	2 Illumination Patterns		3 Illumination Patterns		4 Illumination Patterns		8 Illumination Patterns	
	Random	Learned	Random	Learned	Random	Learned	Random	Learned
MNIST	14 ± 6	28 ± 9	20 ± 11	75 ± 19	32 ± 14	102 ± 10	61 ± 19	113 ± 11
CelebA	13 ± 2	19 ± 3	14 ± 4	28 ± 2	23 ± 5	81 ± 4	43 ± 8	98 ± 15



(a) MNIST **(b)** CelebA

Fig. 1: 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.

Table 2: Reconstruction PSNR (`mean` $\pm$ `std`) of different algorithms using random patterns and our learned patterns. The number of patterns is 4 in each case. Here we round the PSNR values to integers to fit the width of the page. *For Deep Model [30] experiments, patterns are normalized to $[-1, 1]$ range. **For Deep Model, the image size for CelebA generator is 64×64.

	MNIST		CelebA	
	Random	Learned	Random	Learned
HIO [31]	16 ± 9	37 ± 19	38 ± 5	102 ± 5
GS [32]	16 ± 9	37 ± 19	38 ± 4	102 ± 5
WirtFlow [12]	22 ± 16	48 ± 25	20 ± 2	39 ± 3
AmpFlow [26]	42 ± 32	74 ± 48	42 ± 8	138 ± 11
PhaseMax [27]	14 ± 4	24 ± 8	32 ± 2	148 ± 2
Deep Model [30]*	31 ± 2	32 ± 3	22 ± 3**	23 ± 2**
Ours - K=50	32 ± 14	102 ± 10	23 ± 5	81 ± 4
Ours - K=100	51 ± 19	186 ± 15	33 ± 4	132 ± 7

trained our DCGAN model for CelebA. 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. The reconstruction results for the Deep Model also directly depend on the quality of the trained generative models. In our experiments, we were not able to generate images with PSNR higher than 30dB using the generative models.

For the case of Random illumination, we selected the best PSNR from 5 independent trials and report the average computation time for each experiment. In all the cases, we tuned the parameters that provide best results.

The reconstruction PSNR (in dB) is reported in Table 2. We observe that our proposed method with learned patterns performs significantly better than all other algorithms in reconstruction quality.

An interesting attribute of our learned patterns is that they can be used with different algorithms. We observe in Table 2 that our learned patterns provide better results compared to Random patterns with all the phase retrieval algorithms, even though the patterns were not optimized for those algorithms.

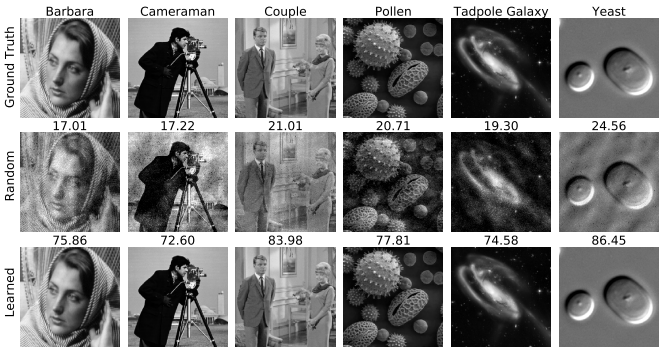


Fig. 2: First Row: Ground truth images from image processing standard test datasets. **Second Row:** Reconstruction using random illumination patterns with uniform random distribution [0, 1] (we selected $T = 4$ patterns that provided best results on celebA test images in **30 trials**). PSNR numbers 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 several classical images. Some results are shown in Fig. 2. 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 supports the generalizability of our learned illumination patterns.

5. CONCLUSION

We presented a framework to learn the illumination patterns for coded diffraction imaging by formulating an iterative phase retrieval algorithm as a fixed unrolled network. It only takes a small number of training images to achieve near-perfect reconstruction whereas random patterns fail. In addition, the learning process of our illumination patterns is highly data efficient and requires a small number of training samples. The learned patterns generalize to different datasets and algorithms that were not used during training.

6. 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] 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.

[4] L. Yeh, J. Dong, J.and Zhong, L. Tian, M. Chen, G. Tang, M. Soltanolkotabi, and L. Waller, "Experimental robustness of fourier ptychography phase retrieval algorithms," *Optics express*, vol. 23, no. 26, pp. 33214–33240, 2015.

[5] C. A Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, "prDeep: Robust phase retrieval with a flexible deep network," in *International Conference on Machine Learning*, 2018, pp. 3501–3510.

[6] 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.

[7] 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.

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

[9] R. Hyder, Viraj S., C. Hegde, and M.S. 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] T. Goldstein and C. Studer, "Phasemax: Convex phase retrieval via basis pursuit," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2675–2689, 2018.

[15] A. S Ghods, Ra.and Lan, T. Goldstein, and C. Studer, "Phaselin: Linear phase retrieval," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–6.

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

[17] 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.

[18] D. V. 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.

[19] Emrah Bostan, Ulugbek S Kamilov, and Laura Waller, "Learning-based image reconstruction via parallel proximal algorithm," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 989–993, 2018.

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

[21] R. Hyder, Z. Cai, and M. S. Asif, "Solving phase retrieval with a learned reference," in *Proc. European Conf. Comp. Vision (ECCV)*, 2020.

[22] S. Wu, A. Dimakis, F. Sanghavi, S.and 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.

[23] 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.

[24] 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.

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

[26] 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. (NeurIPS)*, 2015, pp. 739–747.

[27] S. Bahmani and J. Romberg, "Phase retrieval meets statistical learning theory: A flexible convex relaxation," in *Artificial Intelligence and Statistics*, 2017, pp. 252–260.

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

[29] 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. (NeurIPS)*.

[30] C. A. Metzler and G. Wetzstein, "Deep S^3PR: Simultaneous source separation and phase retrieval using deep generative models," *arXiv preprint arXiv:2002.05856*, 2020.

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

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