ALTERNATING PHASE PROJECTED GRADIENT DESCENT WITH GENERATIVE PRIORS FOR SOLVING COMPRESSIVE PHASE RETRIEVAL

Rakib Hyder[1]*, Viraj Shah*[2]*, Chinmay Hegde*[2]*, and M. Salman Asif*[1]

[1] Electrical and Computer Engineering , University of California, Riverside
[2] Electrical and Computer Engineering, Iowa State University

ABSTRACT

The classical problem of phase retrieval arises in various signal acquisition systems. Due to the ill-posed nature of the problem, the solution requires assumptions on the structure of the signal. In the last several years, sparsity and support-based priors have been leveraged successfully to solve this problem. In this work, we propose replacing the sparsity/support priors with generative priors and propose two algorithms to solve the phase retrieval problem. Our proposed algorithms combine the ideas from AltMin approach for non-convex sparse phase retrieval and projected gradient descent approach for solving linear inverse problems using generative priors. We empirically show that the performance of our method with projected gradient descent is superior to the existing approach for solving phase retrieval under generative priors. We support our method with an analysis of sample complexity with Gaussian measurements.

***Index Terms*—** Phase retrieval, compressive sensing, inverse problem, generative prior

1. INTRODUCTION

1.1. Motivation

The classical problem of phase retrieval arises in numerous imaging applications [1, 2], where only the magnitude of the light rays can be measured but not the phase. As each linear observation loses its phase, the highly non-linear forward model makes it challenging to recover the underlying signal. The phase retrieval problem seeks to recover a real- or complex-valued unknown signal $\mathbf{x} \in \mathbb{R}^n$ from its (possibly noisy) amplitude-only observations $\mathbf{y} \in \mathbb{R}^m$ of the form:

$$y_i = |\langle \mathbf{a}_i, \mathbf{x}^* \rangle| + e_i, \quad i = 1, ..., m, \tag{1}$$

We construct $\mathbf{A} = [\mathbf{a_1}\ \mathbf{a_2}\ ...\ \mathbf{a_m}]^T$ with i.i.d. Gaussian entries. For simplicity, we ignore the noise e_i. We consider a setting with $m < n$, thus in general, the inverse problem in (1) is highly ill-posed.

In general, infinitely many possible solutions exist for (1). A conventional approach for solving such a problem is by constraining the solution to a set $\mathcal{M} \subseteq \mathbb{R}^n$ that captures some sort of known structure that $\mathbf{x}^*$ is expected to obey. The resulting optimization can be written as

$$\widehat{\mathbf{x}} = \arg\min\ f(\mathbf{y}; |\mathbf{A}\mathbf{x}|) \quad \text{s.t.}\ \ \mathbf{x} \in \mathcal{M}. \tag{2}$$

RH and MA were supported by the DARPA REVEAL Program and an equipment donation from NVIDIA Corporation. VS and CH were supported in part by grants from NSF CCF-1815101, a Faculty Fellowship from the Black and Veatch Foundation, and an equipment donation from NVIDIA Corporation.

A common modeling assumption on $\mathbf{x}^*$ is *sparsity*, which alleviates the ill-posed nature of the inverse problem, and in fact, makes the accurate recovery of $\mathbf{x}^*$ possible.

However, while being powerful from a computational standpoint, the sparsity prior has somewhat limited discriminatory capability, and it is certainly true that nature exhibits far richer nonlinear structure than sparsity alone. Thus, we focus on a newly emerging family of priors that are *learned* from massive amounts of training data using generative networks such as GAN [3]. A well-trained generator closely captures the notion of a signal (or image) being 'natural' [4]. While such generative priors have been used successfully in solving compressive sensing and other inverse problems [5], including phase retrieval [6, 7], the optimal way to search for the solution within the range of generative prior has not yet been understood well. Most of these methods rely on loss minimization through gradient descent that often fails to search the entire solution space resulting in sub-optimal results. In this work, we provide two algorithms that enable an improved way of searching the solution space. Our work improves on the results of [6, 7] empirically, along with providing mathematical analysis of convergence.

1.2. Our contributions

In this paper, we propose and analyze two phase retrieval algorithms: alternating phase gradient descent (APGD), and alternating phase projected gradient descent (APPGD) to leverage generative priors. We improve over the approaches of [6, 7] by combining the gradient descent and projected gradient descent methods for generative priors [5, 8] with AltMin-based non-convex optimization techniques used in sparse phase retrieval [9, 10].

We adopt a setting similar to [6, 7], and assume that the generator network (say, G) well approximates the high-dimensional probability distribution of the set $\mathcal{M}$, *i.e.*, we expect that for each vector $\mathbf{x}^*$ in $\mathcal{M}$, there exists a vector $\mathbf{x} = G(\mathbf{z})$ that is very close to $\mathbf{x}^*$ in the support of the distribution defined by G.

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = G(\mathbf{z}) \text{ for some } \mathbf{z} \in \mathbb{R}^k\},$$

With this assumption, the solution to (2) can be obtained by solving the following optimization problem:

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - |\mathbf{A}\mathbf{x}|\|^2 \quad \text{s.t.}\ \ \mathbf{x} = G(\mathbf{z}), \tag{3}$$

where $\mathbf{z}$ is the latent code corresponding to image $\mathbf{x}$. Unless otherwise stated, all norms represented by $\|\cdot\|$ in this paper are Euclidean norms.

Recent work in [6, 7] minimizes the objective in (3) directly over the latent variable $\mathbf{z}$ using gradient descent, and sets $\widehat{\mathbf{x}}$ as:

$$\widehat{\mathbf{x}} = G(\arg\min_{\mathbf{z}} \|\mathbf{y} - |\mathbf{A}G(\mathbf{z})|\|^2). \tag{4}$$

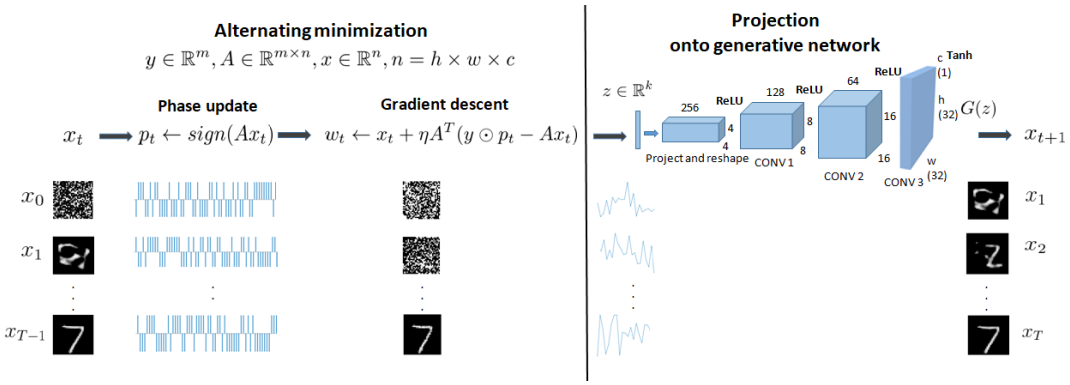


Fig. 1: *Illustration of APPGD algorithm. It has two major steps: alternating minimization and projection onto the range of the generator nework. In alternating minimization step, we update phase and perform one gradient descent update using the updated phase. Starting from a random vector, we perform phase update, gradient descent update step and projection step iteratively to reach the final estimate.*

We refer to this approach as the "gradient descent approach". Given that the generative models usually exhibit highly non-linear behavior, the above objective is highly non-convex. Moreover, direct application of gradient descent over $\mathbf{z}$ limits the explorable solution space, as at any stage it is not possible to explore the region outside the range of the generator. If initialized incorrectly, gradient descent can get stuck in local minima. In practice such algorithms require several restarts in order to provide good performance.

In phase retrieval problems, knowledge of phase and the signal is interdependent, as given the phaseless measurements just knowing the phase often enables us to estimate the signal. Thus, as alternative for solving (3), we can convert the phase retrieval problem to a linear inverse problem by initializing with a random phase $\mathbf{p}$ and update the phase with the solution of the linear inverse problem. Equation (5) describes this approach for the t^{th} iteration.

$$\widehat{\mathbf{x}}_{t+1} = G(\arg\min_{\mathbf{z}} \|\mathbf{p}_t \odot \mathbf{y} - \mathbf{A}G(\mathbf{z})\|^2) \tag{5}$$

We refer this approach as the alternating phase gradient descent (APGD) approach.

We propose a third approach, in which we use projected gradient descent (PGD) to solve (5) directly in the ambient space based on [8]. Through iterative projections, we are able to mitigate the effects of local minima and are able to explore the space outside the range of the generator (G). In PGD, we update our estimate of $\mathbf{x}$ with the standard gradient descent update rule, followed by projection of the output onto the span of generator, G. We refer this approach as the alternating phase projected gradient descent (APPGD) approach. We provide theoretical analysis of our methods, along with extensive experimental results.

1.3. Prior work

Approaches for solving the phase retrieval problem can be broadly classified into convex and non-convex approaches. Convex approaches usually consist of solving a constraint optimization problem after linearizing the problem. The PhaseLift algorithm [11] and its variations [12], [13] come under this category. Typical non-convex approaches include approaches based on Amplitude flow [14, 15] and Wirtinger flow [16, 17, 18, 19].

In recent work, phase retrieval for the cases where underlying signal is sparse is of growing interest. Some of the convex approaches for sparse phase retrieval include [20, 21, 22, 23]. Similarly, non-convex approaches for sparse phase retrieval includes [9, 19, 14]. Our alternating minimization (AltMin)-based approach in this paper is mainly inspired from the non-convex sparse phase retrieval framework advocated in [9, 10].

Different plug-and-play priors [24, 25, 26] have also been proposed to leverage the effect of off-the-shelf image denoisers for solving inverse problems. Recently, various researchers have explored the idea of replacing the sparsity priors with generative priors for solving inverse problems. [5, 8] provided gradient descent and PGD algorithms respectively to solve compressive sensing problem. We use these approaches for the signal estimation in our algorithm. [6, 7] solves the phase retrieval problem using generative priors through enforcing the prior directly by minimizing an empirical risk objective over the domain of the generator. In this paper, we improve over their idea by providing alternative approaches based on AltMin and projected gradient descent.

2. ALGORITHM

In this section we describe the APPGD approach in details. At first, we train a generator $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$ that maps a latent vector $\mathbf{z} \in \mathbb{R}^k$ to a high dimensional sample space $G(\mathbf{z}) \in \mathbb{R}^n$. We assume that our generator network can closely approximate the probability distribution of the set of natural images, $\mathcal{M}$ to which our original images $\mathbf{x}$ belong. With this assumption, we can limit our search for $\widehat{\mathbf{x}}$ only to the range of the generator function, $\mathcal{M}$. The generator G is assumed to be differentiable, and hence we use back-propagation for calculating the gradients of the loss functions involving G for gradient descent updates.

In each iteration of the APPGD algorithm (Alg. 1), three steps are performed: a phase update step, a gradient descent update step, and a projection step.

2.1. Phase update

The first step is to calculate the phase of $\mathbf{Ax}$. For real $\mathbf{A}$ and $\mathbf{x}$, at the t^{th} iteration, we update the phase estimate:

$$\mathbf{p}_t = \text{phase}(\mathbf{A}\mathbf{x}_t) := \text{sign}\left(\mathbf{A}\mathbf{x}_t\right).$$

After calculating the phase vector $\mathbf{p}$, we can use an element-wise product between $\mathbf{p}$ and $\mathbf{y}$ as an estimate of linear measurements and convert the phase retrieval problem into a linear inverse problem.

Algorithm 1 APPGD

1: **Inputs:** $\mathbf{y}$, $\mathbf{A}$, G, T, **Output:** $\widehat{\mathbf{x}}$
2: Choose an initial point $\mathbf{x}_0 \in \mathbb{R}^n$
3: **for** t = 1,... T **do**
4: $\quad \mathbf{p}_{t-1} \leftarrow \text{sign}\left(\mathbf{A}\mathbf{x}_{t-1}\right)$
5: $\quad \mathbf{w}_{t-1} \leftarrow \mathbf{x}_{t-1} + \eta \mathbf{A}^T(\mathbf{y} \odot \mathbf{p}_{t-1} - \mathbf{A}\mathbf{x}_{t-1})$
6: $\quad \mathbf{x}_t \leftarrow \mathcal{P}_G(\mathbf{w}_{t-1}) = G\left(\arg\min_{\mathbf{z}} \|\mathbf{w}_{t-1} - G(\mathbf{z})\|\right)$
7: **end for**
8: $\widehat{\mathbf{x}} \leftarrow \mathbf{x}_T$

2.2. Gradient descent update

The second step is simply an application of a gradient descent update rule on the loss function $f(\cdot)$ which is given as:

$$f(\mathbf{x}) \coloneqq \|\mathbf{y} \odot \mathbf{p} - \mathbf{A}\mathbf{x}\|^2.$$

Thus, the gradient descent update at the t^{th} iteration is given by:

$$\mathbf{w}_t \leftarrow \mathbf{x}_t + \eta \mathbf{A}^T(\mathbf{y} \odot \mathbf{p}_t - \mathbf{A}\mathbf{x}_t),$$

where η is the learning rate.

2.3. Projection step

In projection step, we aim to find an image from the span of the generator, $\mathcal{M}$ which is closest to our current estimate $\mathbf{w}_t$. We define the projection operator $\mathcal{P}_G$ as follows:

$$\mathcal{P}_G\left(\mathbf{w}_t\right) \coloneqq G\left(\arg\min_{\mathbf{z}} L_{in}(\mathbf{z})\right),$$

where L_{in} is the inner loss function defined as,

$$L_{in}(\mathbf{z}) \coloneqq \|\mathbf{w}_t - G(\mathbf{z})\|.$$

We solve the inner optimization problem by running gradient descent with T_{in} number of updates on $L_{in}(\mathbf{z})$. The learning rate η_{in} is chosen empirically for this inner optimization.

In each of the T iterations, we run T_{in} updates for calculating the projection. Therefore, $T \times T_{in}$ is the total number of gradient descent updates required in our approach.

2.4. Analysis

This part of the algorithm is described in Lines 3-7 of Algorithm 1. We can prove that *provided a good initial estimate ($\mathbf{x}_0$), the above algorithm (APPGD) provably converges to* $\mathbf{x}^*$.

The intuition is as follows. Ignoring the noise, the observation model in (1) can be restated as follows:

$$\text{sign}\left(\langle \mathbf{a}_i, \mathbf{x}^* \rangle\right) \odot y_i = \langle \mathbf{a}_i, \mathbf{x}^* \rangle,$$

for all $i = \{1, 2, \ldots, m\}$. To ease notation, denote the *phase vector* $\mathbf{p} \in \mathbb{R}^m$ as a vector that contains the unknown signs of the measurements, i.e., $p_i = \text{sign}\left(\langle \mathbf{a}_i, \mathbf{x} \rangle\right)$ for all $i = \{1, 2, \ldots, m\}$. Let $\mathbf{p}^*$ denote the true phase vector and let $\mathcal{P}$ denote the set of all phase vectors, i.e. $\mathcal{P} = \{\mathbf{p} : p_i = \pm 1, \forall i\}$. Then our measurement model gets modified as:

$$\mathbf{p}^* \odot \mathbf{y} = \mathbf{A}\mathbf{x}^*.$$

Therefore, the recovery of $\mathbf{x}^*$ can be posed as a (non-convex) optimization problem:

$$\min_{\mathbf{x} \in \mathcal{M}, \mathbf{p} \in \mathcal{P}} \|\mathbf{A}\mathbf{x} - \mathbf{p} \odot \mathbf{y}\| \tag{6}$$

To solve this problem, we alternate between estimating $\mathbf{p}$ and $\mathbf{x}$. We perform two estimation steps:

(a) if we fix the signal estimate $\mathbf{x}$, then the minimizer $\mathbf{p} \in \mathcal{P}$ is given in closed form as:

$$\mathbf{p} = \text{sign}\left(\mathbf{A}\mathbf{x}\right), \tag{7}$$

(b) and if we fix the phase vector $\mathbf{p}$, the signal vector $\mathbf{x} \in \mathcal{M}$ can be obtained by solving:

$$\min_{\mathbf{x} \in \mathcal{M}} \|\mathbf{A}\mathbf{x} - \mathbf{p} \odot \mathbf{y}\|_2. \tag{8}$$

We now analyze our proposed descent scheme. We obtain:

Theorem 2.1. *Suppose we have an initialization $\mathbf{x}_0 \in \mathcal{M}$ satisfying $\text{dist}\left(\mathbf{x}_0, \mathbf{x}^*\right) \leq \delta_0 \|\mathbf{x}^*\|_2$, for $0 < \delta_0 < 1$, and suppose the number of (Gaussian) measurements,*

$$m > C\left(kd \log n\right),$$

for some large enough constant C. Then with high probability the iterates $\mathbf{x}_{t+1}$ of Algorithm 1, satisfy:

$$\text{dist}\left(\mathbf{x}_{t+1}, \mathbf{x}^*\right) \leq \rho\, \text{dist}\left(\mathbf{x}_t, \mathbf{x}^*\right), \tag{9}$$

where $\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}^* \in \mathcal{M}$, and $0 < \rho < 1$ is a constant.

Proof sketch: The high level idea behind the proof is that with a δ-ball around the true signal $\mathbf{x}^*$, the "phase noise" can be suitably bounded in terms of a constant times the signal estimation error. To be more precise, suppose that $\mathbf{z}^* = \mathbf{A}\mathbf{x}^* = \mathbf{p}^* \odot \mathbf{y}$. Then, at any iteration t, we have:

$$\begin{aligned}\mathbf{z}_t &= \mathbf{p}_t \odot \mathbf{y} \\ &= \mathbf{p}^* \odot \mathbf{y} + (\mathbf{p}_t - \mathbf{p}^*) \odot \mathbf{y} \\ &= \mathbf{z}^* + \mathbf{e}_t,\end{aligned}$$

where $\mathbf{e}_t$ can be viewed as the "phase noise". Now, examining Line 6 of the above algorithm, we have that $\mathbf{x}_t$ is the output of APPGD after t iterations. An "unpacking" argument similar to the one in [10] indicates that:

$$\|\mathbf{x}_t - \mathbf{x}^*\| \leq \alpha \|\mathbf{x}_{t-1} - \mathbf{x}^*\| + \beta \|\mathbf{e}_t\|,$$

where α is a small enough constant. We will show that $\|\mathbf{e}_t\|$ can be also bounded in terms of $\|\mathbf{x}_{t-1} - \mathbf{x}^*\|$, via Lemma 2.2 below. Consequently:

$$\|\mathbf{x}_t - \mathbf{x}^*\| \leq \rho \|\mathbf{x}_{t-1} - \mathbf{x}^*\|,$$

where ρ is a small enough constant.

We therefore achieve a per-step error reduction scheme if the initial estimate x_0 satisfies $\|x_0 - \mathbf{x}^*\| \leq \delta_0 \|\mathbf{x}^*\|$. This result can be trivially extended to the case where the initial estimate x_0 satisfies $\|x_0 + \mathbf{x}^*\| \leq \delta_0 \|\mathbf{x}^*\|$, hence giving the convergence criterion of the form (for $\rho < 1$):

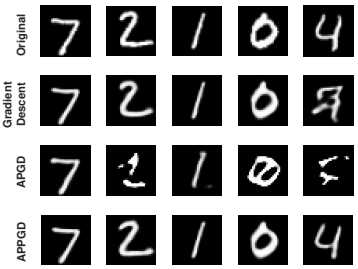
$$\text{dist}\left(\mathbf{x}_t, \mathbf{x}^*\right) \leq \rho\, \text{dist}\left(\mathbf{x}_{t-1}, \mathbf{x}^*\right).$$

We now state Lemma 2.2 without proof. A proof will be provided in an extended version of this paper. The proof is an adaptation of the seminal analysis of [5].

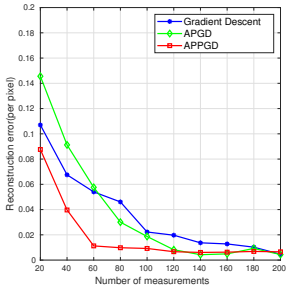
Lemma 2.2. *Suppose that the generator network model $G(\cdot)$ is comprised of d layers of neurons with ReLU activation functions and weight matrices with bounded operator norms. As long as the initial estimate is a small distance away from the true signal $\mathbf{x}^* \in \mathcal{M}$ (i.e. $\text{dist}\left(\mathbf{x}_0, \mathbf{x}^*\right) \leq \delta \|\mathbf{x}^*\|$) and subsequently, $\text{dist}\left(x_t, \mathbf{x}^*\right) \leq \delta \|\mathbf{x}^*\|$, where $\mathbf{x}_t$ is the t^{th} update of Algorithm 1, then the following bound holds for any $t \geq 0$:*

$$\|\mathbf{e}_{t+1}\| \leq \rho_1 \|\mathbf{x}_t - \mathbf{x}^*\|,$$

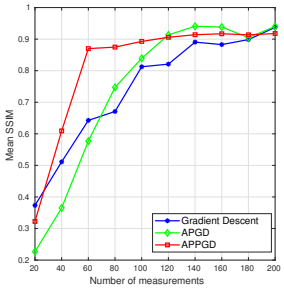
with high probability, as long as $m > C(kd \log n)$ and $\rho_1 < 1$ is a constant.



(a) Reconstruction results on MNIST for three different approaches with m = 60 measurements.

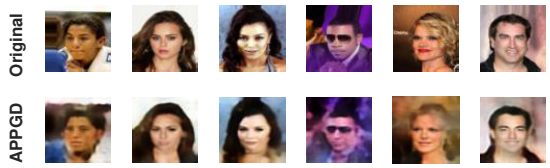


(b) Reconstruction error (per pixel) for three approaches on MNIST.

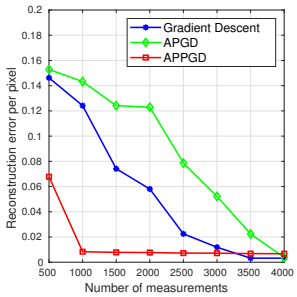


(c) Mean SSIM for three approaches on MNIST.

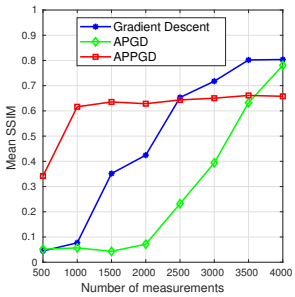
Fig. 2: *Comparison of three approaches on MNIST test set.*



(a) Reconstruction results on celebA dataset for APPGD with m = 1000 measurements.



(b) Reconstruction error (per pixel) for three approaches on celebA.



(c) Mean SSIM for three approaches on celebA.

Fig. 3: *Comparison of three approaches on celebA test set and some reconstruction results for our APPGD algorithm.*

3. MODELS AND EXPERIMENTS

In this section, we describe our experimental setup and report the performance comparisons of the three approaches. We use two different generative models for the MNIST and CelebA datasets. Generative model for celebA follows DCGAN framework [27] except that we do not use any batchnorm layer because gradient through batchnorm layer is dependent on batch size and the distribution of the batch. The generator architecture for MNIST experiments is shown in Fig. 1. We train our generators by jointly optimizing generator parameters, γ and latent code, $\mathbf{z}$ using SGD optimization by following the procedure from [28]. We use squared-loss function, $l_2(\mathbf{x}, \widehat{\mathbf{x}}) = \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$ to train the generators. We choose z from standard normal distribution on $\mathbb{R}^k$ and then rescaled it by its Euclidean norm. We project $\mathbf{z}$ back to the unit norm ball after each gradient update.

In our experiments, we choose the entries of the matrix A independently from a $\mathcal{N}(0, \frac{1}{m})$ distribution. Although we ignore the presence of noise, it is possible to replicate our experiments with additive Gaussian noise. For all the approaches we kept the number of update steps fixed. We do not allow random restarts. For fair comparison, we initialize $\mathbf{x}$ with the same random vector for all the approaches and perform the same sign correction as in [6] on them.

We have our first set of experiments with three different approaches on a generator trained over training set of MNIST dataset resized to 32×32 pixel. Considering that the representation error is very small, we test three approaches on 10 images from the test set of MNIST dataset and provide both quantitative and qualitative results. For APPGD approach, at gradient descent step, we choose $\eta = 0.9$ because we need a meaningful output before passing it to the projection step [8]. We can also perform gradient descent update for couple of times at the first iteration before projecting it onto the range of generator so that we can start from a good initial point. For all three approaches, we use learning rate for z, $\eta_{in} = 0.01$. We use $T = 50$ and $T_{in} = 500$ for APPGD and APGD approaches. For fair comparison, we use 2500 iterations for Gradient descent approach. We use the reconstruction error $= \|\widehat{\mathbf{x}} - \mathbf{x}^*\|^2$ and SSIM for comparison. In Fig. 2b, we show the reconstruction error comparisons and in Fig. 2c, we show SSIM comparison for increasing values of number of measurements. As the input images are not chosen from the span of the generator itself, it is not possible to reach zero error. However, we observe from 2b that APPGD aproach can reach near zero error with only 60 measurements which is significantly less than the other two approaches. Fig. 2a depicts reconstruction results for some of the selected MNIST images for three approaches.

For our second set of experiments, we train a generator over celebA dataset. For training, we resized the celebA dataset composed of 202,599 colored images of celebrity faces to $64 \times 64 \times 3$ and kept $\frac{1}{32}$ of the images apart. We do not use the aligned and cropped version which includes only the faces in the images. We train our generator for CelebA on the rest of the images.

We experiment on a subset of 10 images from the held out test dataset and report results. We set total number of updates to 1500, with $T = 50$ and $T_{in} = 300$ for APGD and APPGD approaches. Learning rates for APPGD are set as $\eta = 0.9$ and $\eta_{in} = 0.3$. Learning rates for APGD and Gradient Descent approaches are set as $\eta_{in} = 0.003$ for their best performance for the fixed total number of updates. Image reconstruction results from $m = 1000$ measurements with APPGD algorithm are displayed in Fig. 3a. We show comparison of three approaches in terms of reconstruction error in 3b and in terms of SSIM in 3c. We observe that APPGD can achieve good reconstruction with much less measurements than the other approaches for celebA.

4. 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] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2014, pp. 2672–2680.

[4] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

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

[6] 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.

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

[8] V. Shah and C. Hegde, "Solving Linear Inverse Problems Using GAN Priors: An Algorithm with Provable Guarantees," *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, 2018.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] Gang Wang, Liang Zhang, Georgios B. Giannakis, Mehmet Akcakaya, and Jie Chen, "Sparse phase retrieval via truncated amplitude flow," *IEEE Trans. Signal Processing*, vol. 66, pp. 479–491, 2018.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] S. Venkatakrishnan, C. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conf. on Signal and Inf. Processing*. IEEE, 2013, pp. 945–948.

[25] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Processing*, vol. 28, no. 3, pp. 1220–1234, 2019.

[26] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, 2017, pp. 3929–3938.

[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Proc. Int. Conf. Learning Representations (ICLR)*, 2016.

[28] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the latent space of generative networks," in *Proc. Int. Conf. Machine Learning*, 2018.