Optimizing Intermediate Representations of Generative Models for Phase Retrieval

Tobias Uelwer[*]
Department of Computer Science
Technical University of Dortmund
tobias.uelwer@tu-dortmund.de

Sebastian Konietzny[*]
Department of Computer Science
Technical University of Dortmund
sebastian.konietzny@tu-dortmund.de

Stefan Harmeling
Department of Computer Science
Technical University of Dortmund
stefan.harmeling@tu-dortmund.de

Abstract

Fourier phase retrieval is the problem of reconstructing images from magnitude-only measurements. It is relevant in many areas of science, e.g., in X-ray crystallography, astronomy, microscopy, array imaging and optics. When training data is available, generative models can be used to constrain the solution set. However, not all possible solutions are within the range of the generator. Instead, they are represented with some error. To reduce this representation error in the context of phase retrieval, we first leverage a novel variation of intermediate layer optimization (ILO) to extend the range of the generator while still producing images consistent with the training data. Second, we introduce new initialization schemes that further improve the quality of the reconstruction. With extensive experiments, we can show the benefits of our modified ILO and the new initialization schemes.

1 Introduction and Related Work

Recently, generative models have shown impressive performance to regularize inverse problems. However, the solutions is required to lay in the range of the generator. To decrease the representation error and thereby boost the expressiveness of these generative models, the idea of optimizing intermediate representations has been applied to various linear inverse problems [1; 2].

In our paper, we show that this idea can be extended to solve the more difficult non-linear inverse problem of phase retrieval. In particular, we tackle the challenging problem of non-oversampled Fourier phase retrieval. For brevity, we define the problem for one dimension.

Fourier Phase Retrieval: In many imaging applications in physics, we are only able to measure the magnitude of the Fourier transform of an image, e.g., in X-ray crystallography [3], astronomy [4], microscopy [5], array imaging [6] or optics [7]. The magnitude measurements are given as

$$y = |\mathcal{F}x|, \tag{1}$$

where $\mathcal{F}$ is the discrete Fourier transform and $x \in \mathbb{R}^n$ is the original image (here represented as a vector). As the problem is highly underdetermined we require additional information about x, which, in this paper, will be learned from example images to constrain the solution space. We assume that the Fourier measurements are not oversampled, i.e., we consider the case, where the image x has

[*] equal contribution

NeurIPS 2022 AI for Science Workshop.

not been zero-padded which corresponds to a practical more relevant setup. A related problem is the Gaussian phase retrieval problem, where $\mathcal{F}$ in Equation 1 is replaced by a (not necessarily square) matrix A that has normally-distributed entries.

Note, that the problem of phase retrieval are more difficult than linear inverse problems due to the non-linearity (the absolute value) in the forward model. In the following, we give a short overview of the various approaches distinguishing between optimization-based and learning-based methods.

1.1 Methods without Learning for Phase Retrieval

One of the first (Fourier) phase retrieval algorithms is the Gerchberg-Saxton (GS) method [8] that is based on alternating projections. Fienup [9] later introduced the hybrid-input-output (HIO) algorithm which improved the Gerchberg-Saxton algorithm. The Gaussian phase retrieval problem was approached by Candes et al. [10] who used methods based on Wirtinger derivatives to minimize a least-squares loss. Wang et al. [11] considered a similar idea but used a different loss function and so-called truncated generalized gradient iterations. Holographic phase retrieval is a related problem, which aims to reconstruct images from magnitude measurements where a known reference is assumed to be added onto the image. Lawrence et al. [12] approached this problem using untrained neural-network priors.

1.2 Learning-based Methods for Phase Retrieval

Recently, learning-based methods for solving phase retrieval problems gained a lot of momentum. They can be classified into two groups: supervised methods and unsupervised methods.

Supervised methods: These methods directly learn a mapping that reconstructs images from the measurements. Learning feed-forward networks for solving Fourier phase retrieval was done by Nishizaki et al. [13]. This approach was later extended to a neural network cascade by Uelwer et al. [14], who also used conditional generative adversarial networks to solve various phase retrieval problems [15]. Supervised learning of reference images for holographic phase retrieval was done by Hyder et al. [16] and Uelwer et al. [17] gave further insights.

Unsupervised methods: Unsupervised methods are agnostic with respect to the measurement operator, because they are trained on a dataset of images without the corresponding measurements. Generative priors for compressive phase retrieval have been analyzed by Hand et al. [18] and Liu et al. [19]. Alternating updates for compressive phase retrieval with generative models was proposed by Hyder et al. [20]. Manekar et al. [21] used a passive loss formulation to tackle the non-oversampled Fourier phase retrieval problem.

In the context of these related works, the method that we propose in this paper can be seen as an unsupervised learning-based approach that can optionally be extended with a supervised component, as we detail in Section 2.2.

1.3 Optimizing Representations of Generators for Inverse Problems

Generative models have been used to solve linear inverse problems, e.g., compressed sensing [22], image inpainting [23]. To decrease the representation error and thereby boost the expressiveness of these generative models, the idea of optimizing intermediate representations was successfully applied in the context of compressed sensing [1], image inpainting and super-resolution [2]. The former paper called this approach generator surgery, the latter intermediate layer optimization (ILO).

1.4 Our Contributions

The contributions of this paper are the following:

1. We reformulate and extend the idea of optimizing intermediate representations in generative models for solving different phase retrieval problems. In particular, we propose a subsequent optimization that further improves the reconstruction.
2. We introduce new learned and non-learned initialization schemes for generator-based approaches to solve inverse problems, reducing the need for multiple runs with random restarts.

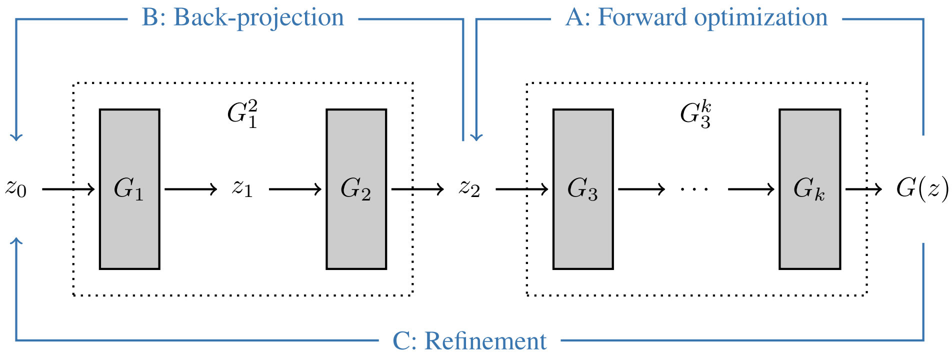

Figure 1: Structure of the generator network and, in blue, the parts relevant for steps A, B and C.

3. In extensive experiments, we consider the underdetermined Fourier phase retrieval and show that our method provides state-of-the-art results.
4. By analyzing various combinations of methods and generator networks, we show the influence of the choice of the generator. In further ablation studies, we show the importance of each component of our method.

2 Phase Retrieval with Generative Models

The basic idea of applying a trained generative model $G : \mathbb{R}^k \to \mathbb{R}^n$ to the phase retrieval problem is to plug $G(z)$ into the least-squares data fitting term and to optimize over the latent variable z, i.e.,

$$\min_z \left\| |\mathcal{F}G(z)| - y \right\|_2^2. \tag{2}$$

This method has been used for Gaussian phase retrieval by Hand et al. [18]. While this approach yields good results, the reconstruction quality is limited by the range of the generator network G. This issue is, for example, also discussed in the work of Asim et al. [24]. In the following, we explain how the range of G can be extended by optimizing intermediate representations of G instead of only solving Equation (2) with respect to the latent variable z.

Notation: For the exposition of the method we use the following notation: The generator network $G = G_k \circ \cdots \circ G_1$ can be written as the concatenation of k layers $G_1, \ldots, G_k$. The subnetwork consisting of layers i through j is denoted by $G_i^j = G_j \circ \cdots \circ G_i$. Note, that $G = G_1^k = G_1^i \circ G_{i+1}^k$ for $1 \leq i \leq k$. The output of layer G_i is written as z_i, i.e., $z_i = G_i(z_{i-1})$. The input of G is z_0.

Furthermore, we define the ℓ_1-ball with radius r around z as $B_r(z) = \left\{ x \mid \|x - z\|_1 \leq r \right\}$.

2.1 Phase Retrieval with Intermediate Layer Optimization (PRILO)

The key idea of ILO is to vary the intermediate representations learned by a sub-network of the generator G while ensuring an overall consistency of the solution. The latter is particularly challenging for non-linear problems like phase retrieval. As we will show in the experiments, additional steps and special initialization schemes are essential to obtain useful results. All of these steps will be explained next.

The generator network captures the prior knowledge and restricts the solutions of the (underdetermined) phase retrieval problem. In this work, we use two important ideas:

Range extension: Hidden representations z_i for $i > 0$ are allowed to vary outside the range of G_1^i. This will reduce the representation error of the generator.

Image consistency: Solutions must be realistic, non-degenerate and similar to the the training dataset. This is achieved by back-projection ensuring that z_i is not too far away from the image of some z_0.

Note that one has to strive for a trade-off between range extension and image consistency, where range extension facilitates the optimization, while image consistency regularizes the solution.

Concretely, we repeatedly split the generator G into sub-networks G_1^i and G_{i+1}^k and iteratively optimize the intermediate representations z_i resulting from G_1^i (instead of simply optimizing the initial latent variable z_0). In this way, we go outside the range of the sub-network G_1^i to reduce the mean-squared magnitude error. To make the resulting image more consistent we back-project z_i closer to the range of G_1^i. We will show that z_0 from the back-projection is not only a good candidate for the overall optimization, but also serves as a good initialization for further optimization leading to a significant refinement of the overall solution.

To get started, we initially optimize the input variable z_0 (minimize Equation (2)). This provides starting points for all intermediate representations z_i for $i > 0$. The overall procedure then iteratively applies the following three steps to different intermediate layers:

A. Forward optimization: vary the intermediate representation z_i to minimize the magnitude error while staying in the ball with radius r_i around the current value of z_i, i.e.,

$$z_i^* = \underset{z \in B_{r_i}(z_i)}{\arg\min} \left\| |\mathcal{F} G_{i+1}^k(z)| - y \right\|_2^2. \tag{3}$$

Note, that this step possibly leaves the range of G_1^i (range extension). By introducing ball-constraints on the optimized variable we avoid overfitting the measurements.

B. Back-projection: find an intermediate representation $G_1^i(z_0)$ that is close to the optimal z_i^* from step A, i.e.,

$$\bar{z}_0 = \underset{z \in B_{s_i}(\mathbf{0})}{\arg\min} \left\| G_1^i(z) - z_i^* \right\|_2^2, \tag{4}$$

where $\mathbf{0} \in \mathbb{R}^l$ denotes the vector of all zeros. By doing so we ensure that there exists a latent z_0 that yields the reconstruction. Thereby, we regularize the solution from step A to obtain image consistency.

While these two steps have been shown to improve the overall performance of linear inverse problems, we found that for phase retrieval the following refinement step significantly improves the reconstructions.

C. Refinement: Our intuition here is that the back-projected latent $\bar{z}_0$ serves as a good initialization for further optimization. Starting from the $\bar{z}_0$ found in step B, we again optimize the measurement error

$$z_0^* = \underset{z \in B_{r_0}(\bar{z}_0)}{\arg\min} \left\| |\mathcal{F} G(z)| - y \right\|_2^2. \tag{5}$$

Note, that as we only optimize the latent variable of G, we do not have to apply the back-projection again for this refinement step.

These steps can be repeated for multiple intermediate representations. After that we obtain the final reconstruction as $x^* = G(z_0^*)$. Steps A, B and C are visualized in Figure 1.

The three optimization problems stated in Equations (3)-(5) are solved using projected gradient descent. This means that after each gradient descent step the iterate is projected back onto the appropriate ℓ_1-ball. In our implementation this projection is implemented using the method described by Duchi et al. [25].

2.2 Initialization Schemes

Due to the difficulty of the phase retrieval problem, many methods are sensitive to the initialization. Candes et al. [10] proposed a spectral initialization technique that is often used for compressive Gaussian phase retrieval. Since it requires one to solve a large eigenvalue problem, this method can be quite costly. A common approach to improve reconstruction performance is to use random restarts and to select the solution with the lowest magnitude error after the optimization. However, this requires running the method multiple times. Instead, we propose two fast initialization schemes that use the generative model and the available magnitude information before the optimization.

Magnitude-Informed Initialization (MII): Deep generative models can generate a lot of images at relatively low cost: instead of optimizing Equation (2) with random restarts, we sample a set of starting points $Z = \{z_0^{(0)}, \ldots, z_0^{(p)}\}$ and select the one with the lowest magnitude mean-squared error (MSE),

$$z_0^{\text{init}} = \underset{z \in Z}{\arg\min} \left\| |\mathcal{F}G(z)| - y \right\|_2^2, \tag{6}$$

before the optimization. The empirical reason for this choice is that the MSE between the unknown target image and the initial reconstructions $G(z)$ for $z \in Z$ strongly correlates with the MSE of their magnitudes (correlation $\rho = 0.91$). This initialization scheme is applicable to other reconstruction algorithms that search in the latent space of a generative model as well.

Learned Initialization (LI): The generator is trained on a dataset of images that are characteristic for the problem. We use the generator to train an encoder network E_θ that maps magnitude measurements y to latent representations z_0. Once the encoder is trained, we can use it to predict an initialization for the optimization discussed in Section 2.1.

A naive way to train the encoder network is to create input/output pairs by starting with random latent vectors z_0 and combining them with the magnitudes $|\mathcal{F}G(z_0)|$ of the corresponding image. The disadvantage is that the original training images are only implicitly used in this approach (because those were used to train the generator). To leverage the generator and also the training images, we estimate the weights θ of the encoder E_θ, such that encoded magnitudes $E_\theta(y)$ generate an image $G(E_\theta(y))$ that is close to the original image x. This idea originates from GAN inversion [26].

More precisely, we minimize a combination of three loss functions to train the encoder E_θ,

$$\mathcal{L}_{\text{MSE}}(G(E_\theta(y)), x) + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}}(G(E_\theta(y), x)) + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}(D_\phi(G(E_\theta(y)), D_\phi(x))), \tag{7}$$

where $\mathcal{L}_{\text{MSE}}$ is the image MSE, $\mathcal{L}_{\text{perc}}$ the LPIPS loss [27], and $\mathcal{L}_{\text{adv}}$ is a Wasserstein adversarial loss [28] with gradient penalty [29] (using the discriminator network D_ϕ). In our experiments, we set $\lambda_{\text{perc}} = 5 \cdot 10^{-5}$ and $\lambda_{\text{adv}} = 0.1$. Note, that the generator network is fixed and we only optimize the encoder weights θ and the discriminator weights ϕ to solve a learning objective.

Additionally, we also found it helpful to apply a small normally-distributed perturbation with mean 0 and standard deviation $\sigma = 0.05$ to the predicted latent representation. We do so because using a fixed initialization leads to a deterministic optimization trajectory when running the optimization multiple times.

For better exploration of the optimization landscape we also apply gradient noise during optimization. In combination with the projection onto the feasible set the update reads as

$$z^{(k+1)} = P\left(z^{(k)} - \alpha\left(\nabla_z f\left(z^{(k)}\right) + u^{(k)}\right)\right), \tag{8}$$

where f is the objective function corresponding to the current step, $u^{(k)} \sim \mathcal{N}(0, \sigma_k^2 I)$ with $\sigma_k^2 = \frac{\eta}{(1+k)^\gamma}$ and P is the projection onto $B_r(x^{(0)})$. In our experiments we set $\eta = 0.02$ and $\gamma = 0.55$. This noise decay schedule was proposed by Welling and Teh [30] and is also discussed by Neelakantan et al. [31] in the context of neural network training.

One drawback of the learned initialization is that it requires one to retrain the encoder network when the measurement matrix changes. In the following, we only evaluate this approach for the Fourier phase retrieval problem.

3 Experimental Evaluation

We evaluate our method on the following datasets: MNIST [32], EMNIST [33], FMNIST [34], and CelebA [35]. The first three datasets consist of 28×28 grayscale images, whereas the latter dataset is a collection of $200,000$ color images that we cropped and rescaled to a resolution of 64×64.

Similar to Hand et al. [18], we use a fully-connected variational autoencoder (VAE) for the MNIST-like datasets and a DCGAN [36] for the CelebA dataset. Both architectures were also used by Bora et al. [22] for compressed sensing. Going beyond existing works, we are interested in improving the performance on the CelebA dataset even further, thus we considered deeper, more expressive

Table 1: Fourier phase retrieval: Our proposed method (being unsupervised) performs best among classical, unsupervised and supervised approaches for MNIST and EMNIST, and second best for FMNIST (in terms of PSNR). Best values are printed **bold** and second-best are underlined.

		MNIST		FMNIST		EMNIST	
	Learning	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)	PSNR (↑)	SSIM (↑)
GS [8]	–	15.1826	0.5504	12.7002	0.3971	13.1157	0.5056
HIO [9]	–	23.4627	0.5274	12.9530	0.4019	14.2230	0.4942
DPR [18]	unsup.	23.9871	0.9005	18.8657	0.6846	20.9448	0.8568
PRILO (ours)	unsup.	42.7584	0.9843	20.2087	0.7133	33.8263	0.9367
PRILO-MII (ours)	unsup.	**43.9541**	**0.9949**	21.4836	0.7560	**37.1807**	**0.9719**
ResNet [13]	sup.	17.0886	0.7292	17.4787	0.6070	14.4627	0.5616
E2E [15]	sup.	18.5041	0.8191	20.2827	0.7400	17.4049	0.7598
CPR [14]	sup.	19.6216	0.8529	20.9064	0.7768	19.3163	0.8537
PRCGAN* [15]	sup.	41.3767	0.9890	**25.5513**	**0.8376**	27.1948	0.9416

Table 2: Fourier phase retrieval on CelebA: Our proposed PRILO combined with LI and StyleGAN achieves the best results in terms of PSNR and SSIM, also against enhanced version of DPR, which uses LI and StyleGAN. Best values are printed **bold** and second-best are underlined.

			CelebA	
	Base model	Learning	PSNR (↑)	SSIM (↑)
GS [8]	–	–	10.4036	0.0637
HIO [9]	–	–	10.4443	0.0510
DPR [18]	DCGAN	unsup.	16.9384	0.4457
DPR-MII	DCGAN	unsup.	17.8651	0.4898
PRILO	DCGAN	unsup.	18.3597	0.5159
PRILO-MII	DCGAN	unsup.	18.5656	0.5264
DPR	Progressive GAN	unsup.	18.4384	0.5276
DPR-MII	Progressive GAN	unsup.	19.5213	0.5738
PRILO	Progressive GAN	unsup.	19.2779	0.5665
PRILO-MII	Progressive GAN	unsup.	19.9057	0.5910
DPR	StyleGAN	unsup.	18.2606	0.4859
DPR-MII	StyleGAN	unsup.	20.3889	0.5972
PRILO	StyleGAN	unsup.	18.5542	0.5143
PRILO-MII	StyleGAN	unsup.	21.4223	0.6358
E2E [15]	–	sup.	20.1266	0.6367
PRCGAN* [15]	–	sup.	22.1951	0.6846
DPR-LI	StyleGAN	sup.	22.6223	0.7021
PRILO-LI	StyleGAN	sup.	**23.3378**	**0.7247**

generators, like the Progressive GAN [37] and the StyleGAN [38]. Since the optimization of the initial latent space for deeper generators is more difficult, we expect significantly better results by adaptively optimizing the successive layers of these models.

3.1 Phase Retrieval with Fourier Measurements

For the problem of Fourier phase retrieval, we compare our method first with the HIO algorithm [9] and the GS algorithm [8] (both having no learning component) and then with the following supervised learning methods: an end-to-end (E2E) learned multi-layer-perceptron [15], a residual network [13], a cascaded multilayer-perceptron [14] and a conditional GAN approach [15]. Additionally, we apply DPR which was originally only tested on Gaussian phase retrieval [18].

Table 1 and Table 2 show the mean peak-signal-to-noise-ratio (PSNR) and the mean structural similarity index measure (SSIM) [39]. Each reported number was calculated on the reconstructions of 1024 test samples. We allow four random restarts and select the generated sample resulting in the lowest measurement error. As one can see, more expressive models result in better reconstructions.

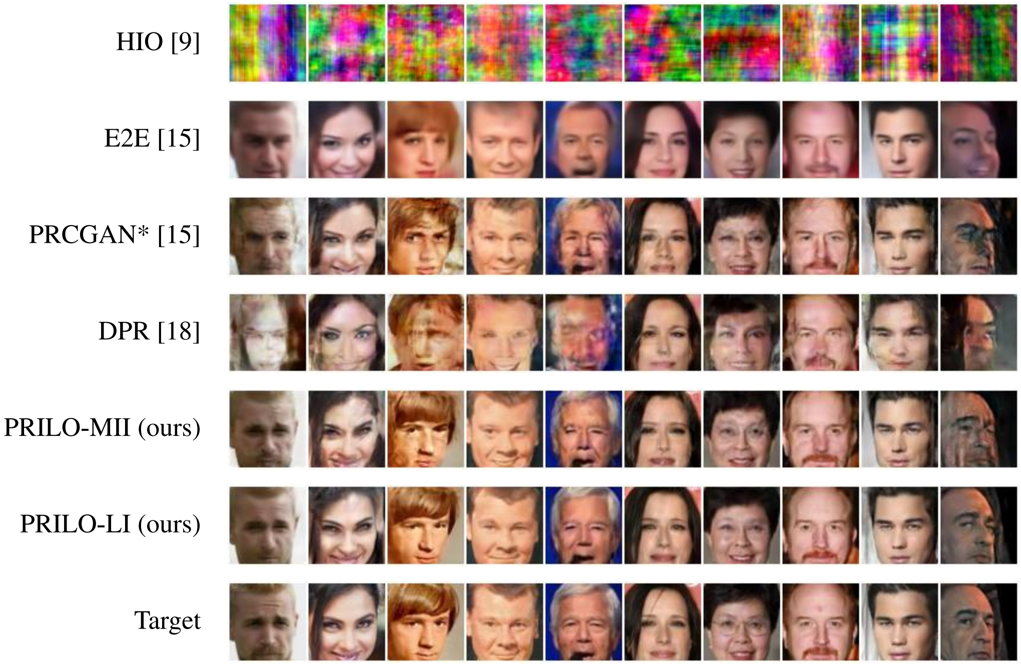

Figure 2: Fourier measurements on CelebA: only PRILO-LI (based on StyleGAN) is able to reconstruct finer details of the faces and produces realistic images with few artifacts.

Furthermore, we note that our proposed initialization schemes, i.e., the LI and MII, lead to improved performance. Figure 2 shows typical reconstructions from the CelebA dataset.

Surprisingly, our unsupervised PRILO-MII approach often quantitatively outperforms the supervised competitors (MNIST and EMNIST). Only for FMNIST we get slightly worse results which we attribute to the performance of the underlying generator of the VAE. Notably, we outperform DPR that uses the same generator network, which shows that the modifications explained in this paper are beneficial. Summarizing, among classical and unsupervised methods our new approach performs best, often even better than the supervised methods.

The CelebA dataset is more challenging and the influence of the choice of the generator networks is substantial. We consider the DCGAN [36], the Progressive GAN [37] and the StyleGAN [38] as base models for our approach (second column in Table 2). Furthermore, the new initialization schemes have a strong impact. For fairness, we also combined the existing DPR approach [18] with the various generator models and the initialization schemes. Thus we are able to study the influence of the different components. Among the generators, StyleGAN performs best. MII improves the results, while LI achieves even better reconstructions. For all previously mentioned combinations, our new method PRILO is better than DPR.

3.2 Ablation Studies

In our experiments, we observed that the performance can be improved by re-running the optimization procedure with a different initialization and selecting the result with the lowest magnitude error which is a common practice in image reconstruction. However, this approach is quite costly. Our initialization schemes described in Section 2.2 can be used to achieve a similar effect without the need of re-running the whole optimization procedure multiple times. In Figure 3, we compare the single randomly initialized latent variable, 5000 initialization using the MII approach and slightly perturbed learned initialization (LI). Again, we consider a test set of 1024 samples. Although, we are still using restarts in combination with our initialization, we observe that already for a single run our initialization outperforms the results of 5 runs.

In order to assess the impact of each step on the reconstruction performance, we perform an ablation study. We compare our complete PRILO-MII and PRILO-LI models with different modified versions

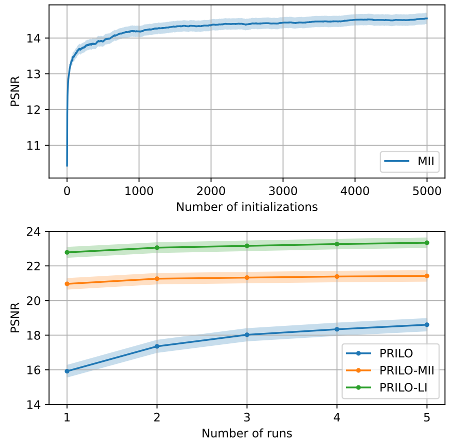

Figure 3: Effect of the number of initializations and number of restarts onto the reconstruction performance. Shaded areas indicate the 95%-confidence interval.

Table 3: Ablation experiments using StyleGAN on CelebA for Fourier measurements.

	PSNR (↑)	SSIM (↑)
PRILO-LI	23.3378	0.7247
only initialization	16.9070	0.4777
only step A	23.0097	0.7108
only step A and B	22.9109	0.7063
no ball constraint	22.8192	0.6974
PRILO-MII	21.4223	0.6358
only initialization	15.0566	0.3565
only step A	20.1736	0.5861
only step A and B	19.9409	0.5719
no ball constraint	19.8461	0.5540

of the model: for one, we consider omitting every optimization, i.e., we only compare with the initialization. Next, we omit the back-projection step (B) and the refinement step (C). We also compare with the variant of our model which we only omit step C. Finally, we analyze the impact of the ball-constraints. Table 3 shows that each of the components is important to reach the results.

4 Conclusion

Generative models play an important role in solving inverse problem. In the context of phase retrieval, we observe that optimizing intermediate representations is essential to obtain excellent reconstructions. Our method PRILO, used in combination with our new initialization schemes, produces better reconstruction results for Fourier phase retrieval than existing (supervised and unsupervised) methods. Notably, in some cases our unsupervised variant PRILO-MII even outperforms existing supervised methods. We also show that the initialization schemes we introduced can easily be adapted to different methods for inverse problems that are based on generative models, e.g., DPR. Our ablation study justifies that each of our used components is essential to achieve the reported results.

Limitations: During our experiments, we observed that some hyperparameter tuning is necessary to achieve good results: the selected intermediate layer highly influences the results and also the radii for the constraints play an important role. Furthermore, our approach is in some cases not able to reconstruct the finer details and is still (to some extend) limited by the generative model.

References

[1] Niklas Smedemark-Margulies, Jung Yeon Park, Max Daniels, Rose Yu, Jan-Willem van de Meent, and Paul Hand. Generator surgery for compressed sensing. *arXiv preprint arXiv:2102.11163*, 2021.

[2] Giannis Daras, Joseph Dean, Ajil Jalal, and Alex Dimakis. Intermediate layer optimization for inverse problems using deep generative models. In *International Conference on Machine Learning*, pages 2421–2432. PMLR, 2021.

[3] Rick P Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

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

[5] Guoan Zheng, Roarke Horstmeyer, and Changhuei Yang. Wide-field, high-resolution fourier ptychographic microscopy. *Nature photonics*, 7(9):739, 2013.

[6] Oliver Bunk, Ana Diaz, Franz Pfeiffer, Christian David, Bernd Schmitt, Dillip K Satapathy, and J Friso Van Der Veen. Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels. *Acta Crystallographica Section A: Foundations of Crystallography*, 63(4):306–314, 2007.

[7] Adriaan Walther. The question of phase retrieval in optics. *Optica Acta: International Journal of Optics*, 10(1):41–49, 1963.

[8] Ralph W Gerchberg. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[9] James R Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

[10] Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

[11] Gang Wang, Georgios B Giannakis, and Yonina C Eldar. Solving systems of random quadratic equations via truncated amplitude flow. *IEEE Transactions on Information Theory*, 64(2): 773–794, 2017.

[12] Hannah Lawrence, David A Barmherzig, Henry Li, Michael Eickenberg, and Marylou Gabrié. Phase retrieval with holography and untrained priors: Tackling the challenges of low-photon nanoscale imaging. *arXiv preprint arXiv:2012.07386*, 2020.

[13] Yohei Nishizaki, Ryoichi Horisaki, Katsuhisa Kitaguchi, Mamoru Saito, and Jun Tanida. Analysis of non-iterative phase retrieval based on machine learning. *Optical Review*, 27 (1):136–141, 2020.

[14] Tobias Uelwer, Tobias Hoffmann, and Stefan Harmeling. Non-iterative phase retrieval with cascaded neural networks. In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 295–306. Springer International Publishing, 2021.

[15] Tobias Uelwer, Alexander Oberstraß, and Stefan Harmeling. Phase retrieval using conditional generative adversarial networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 731–738. IEEE, 2021.

[16] Rakib Hyder, Zikui Cai, and M Salman Asif. Solving phase retrieval with a learned reference. In *European Conference on Computer Vision*, pages 425–441. Springer, 2020.

[17] Tobias Uelwer, Nick Rucks, and Stefan Harmeling. A closer look at reference learning for fourier phase retrieval. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021.

[18] Paul Hand, Oscar Leong, and Vlad Voroninski. Phase retrieval under a generative prior. In *Advances in Neural Information Processing Systems*, pages 9136–9146, 2018.

[19] Zhaoqiang Liu, Subhroshekhar Ghosh, and Jonathan Scarlett. Towards sample-optimal compressive phase retrieval with sparse and generative priors. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] Rakib Hyder, Viraj Shah, Chinmay Hegde, and M Salman Asif. Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7705–7709. IEEE, 2019.

[21] Raunak Manekar, Zhong Zhuang, Kshitij Tayal, Vipin Kumar, and Ju Sun. Deep learning initialized phase retrieval. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020.

[22] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[23] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

[24] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *International Conference on Machine Learning*, pages 399–409. PMLR, 2020.

[25] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[26] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[28] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[29] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. *Advances in neural information processing systems*, 30, 2017.

[30] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[31] Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[34] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[37] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[38] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

A Appendix

A.1 CelebA Reconstructions using the Progressive GAN

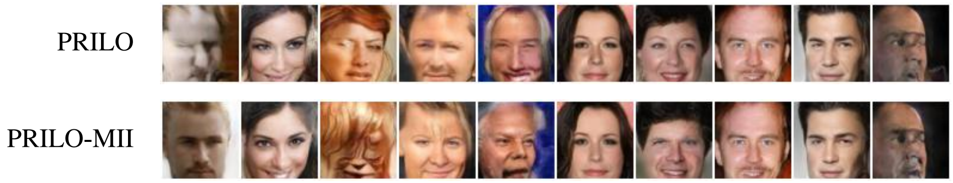

Figure 4: Reconstructions from Fourier measurements on CelebA by PRILO and PRILO-MII based on the Progressive GAN [37].

A.2 CelebA Reconstructions using the DCGAN

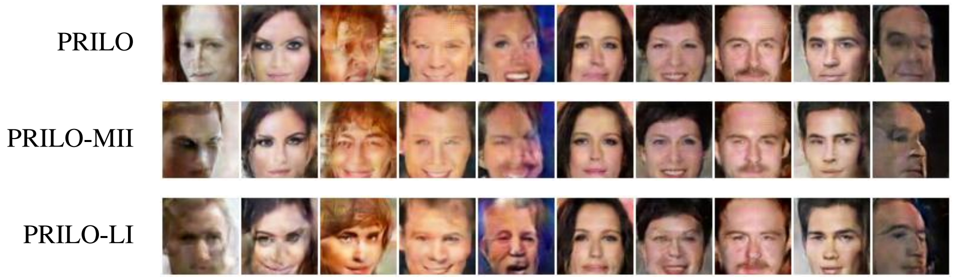

Figure 5: Reconstructions from Fourier measurements on CelebA by PRILO and PRILO-MII based on the DCGAN [36].

A.3 Ablation Experiments: PSNR Throughout the Optimization

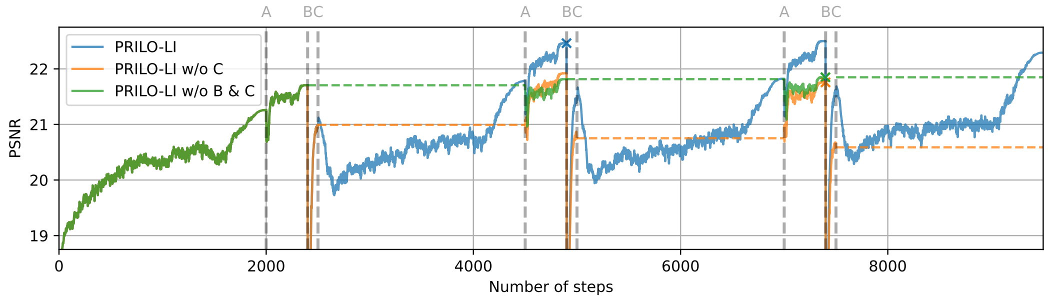

Figure 6: Ablation experiments for PRILO-LI based on StyleGAN on CelebA data: We omit different steps of our method and report the PSNR on a validation set consisting of 64 images. PSNR corresponding to the lowest magnitude error are highlighted.