INVERSE PROBLEMS, DEEP LEARNING, AND SYMMETRY BREAKING

Anonymous authors
Paper under double-blind review

ABSTRACT

In many physical systems, inputs related by intrinsic system symmetries generate the same output. So when inverting such systems, an input is mapped to multiple symmetry-related outputs. This causes fundamental difficulties for tackling these inverse problems by the emerging end-to-end deep learning approach. Taking phase retrieval as an illustrative example, we show that careful symmetry breaking on the training data can help get rid of the difficulties and significantly improve learning performance in real data experiments. We also extract and highlight the underlying mathematical principle of the proposed solution, which is directly applicable to other inverse problems.

1 INTRODUCTION

1.1 INVERSE PROBLEMS AND DEEP LEARNING

For many physical systems, we observe only the output and strive to infer the input. The inference task is often captured by the generic term "inverse problem". Formally, the underlying system is modeled by a forward mapping f, and solving the inverse problem amounts to identifying the inverse mapping f^{-1}. Inverse problems abound in numerous fields and take diverse forms, see, e.g., (Hartley & Zisserman, 2003; Gonzalez & Woods, 2017; Comon, 2010; Colton & Kress, 2013; Herman, 2009; Entekhabi et al., 1994; Ge, 2013). Let $\boldsymbol{y}$ denote the observed output. Traditionally, inverse problems are mostly formulated as regularized optimization problems of the form

$$\min_{\boldsymbol{x}} \; \ell(\boldsymbol{y}, f(\boldsymbol{x})) + \lambda \Omega(\boldsymbol{x}), \tag{1.1}$$

where $\boldsymbol{x}$ represents the input to be estimated, $\ell(\boldsymbol{y}, f(\boldsymbol{x}))$ ensures $\boldsymbol{y} \approx f(\boldsymbol{x})$ (ℓ means loss), $\Omega(\boldsymbol{x})$ encodes prior knowledge about $\boldsymbol{x}$—often added to make the problem well-posed, and λ is a tradeoff parameter. To solve Eq. (1.1) , iterative numerical algorithms are often developed (Kirsch, 2011).

Deep learning has enabled learning data-driven loss ℓ or Ω, or replacing mappings in iterative methods for solving Eq. (1.1) by data-adaptive ones. These ideas can capture structures in practical data not expressible before and tend to lead to faster and/or more effective algorithms. Most radical is perhaps the end-to-end approach: a deep neural network (DNN) is directly set up and trained to approximate the inverse mapping f^{-1}—backed by the famous universal approximation theorem (Poggio et al., 2017) and based on a sufficiently large set of $(\boldsymbol{x}, \boldsymbol{y})$ pairs. Instead of citing the abundance of individual papers, we refer the reader to the excellent review articles (McCann et al., 2017; Lucas et al., 2018; Arridge et al., 2019; Ongie et al., 2020b) on these developments.

1.2 DIFFICULTY WITH SYMMETRIES

In this paper, we focus on the end-to-end learning approach. This approach has recently been widely acclaimed for its remarkable performance on several tasks such as image denoising (Xie et al., 2012), image super-resolution (Dong et al., 2014), image deblurring (Xu et al., 2014), and sparse recovery (Mousavi & Baraniuk, 2017). In these examples, f is linear.

When f is nonlinear, intrinsic symmetries appear in many problems. A couple of quick examples:

- **Fourier phase retrieval (PR)** The forward model is $\boldsymbol{Y} = |\mathcal{F}(\boldsymbol{X})|^2$, where $\boldsymbol{X} \in \mathbb{C}^{n \times n}$ and $\boldsymbol{Y} \in \mathbb{R}^{m \times m}$ are matrices and $\mathcal{F}$ is the 2D (oversampled) Fourier transform. The operation

$|\cdot|$ takes elementwise complex magnitudes. It is well known that translations to the nonzero part of $\boldsymbol{X}$ (if feasible), conjugate flipping of $\boldsymbol{X}$, and global phase transfer $e^{i\theta}\boldsymbol{X}$ for any $\theta \in [0, 2\pi)$ all lead to the same $\boldsymbol{Y}$ (Bendory et al., 2017).

- **Blind deconvolution** The forward model is $\boldsymbol{y} = \boldsymbol{a} \circledast \boldsymbol{x}$, where $\boldsymbol{a}$ is the convolution kernel, $\boldsymbol{x}$ is the signal (e.g., image) of interest, and $\circledast$ denotes the circular convolution. Both $\boldsymbol{a}$ and $\boldsymbol{x}$ are inputs. Here, $\boldsymbol{a} \circledast \boldsymbol{x} = (\lambda \boldsymbol{a}) \circledast (\boldsymbol{x}/\lambda)$ for any $\lambda \neq 0$, and circularly shifting $\boldsymbol{a}$ to the left and shifting $\boldsymbol{x}$ to the right by the same amount does not change $\boldsymbol{y}$ (Lam & Goodman, 2000; Tonellot & Broadhead, 2010)

Solving these inverse problems means recovering the input up to the intrinsic system symmetries, as evidently this is the best one can hope for.

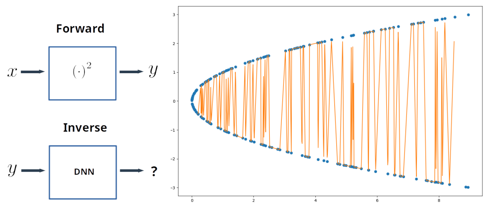

Figure 1: Learn to take square root. (Left) The forward and inverse models; (Right) The function (in orange) determined by the training points.

Symmetries can cause significant difficulty for the end-to-end approach. To see this, suppose we randomly sample real values x_i's and form a training set $\{x_i, x_i^2\}$ and try to learn the square-root function, allowing both positive and negative outputs, using the end-to-end approach. Now if we think of the function determined by the training set, which the neural network is trying to approximate, it is highly oscillatory (see Fig. 1)[1]: the sign symmetry dictates that in the training set, there are frequent cases where x_i^2 and x_j^2 are close but x_i and x_j have different signs and are far apart. Although in theory neural networks with adequate capacity are universal function approximators, in practice they will struggle to learn such irregular functions. For general inverse problems, so long as the forward symmetries can relate remote inputs to the same output, similar problems can surface.

1.3 Our contribution: symmetry breaking

An easy fix to the above issue is fixing all signs of x_i's to be positive (or negative), which we call "symmetry breaking". We generalize this and

- Take phase retrieval (PR) as an example to show how symmetry breaking can be performed and how this can lead to substantial gain in performance. For PR, our algorithm solves the problem in a regime not accessible by previous methods.
- Identify the basic principle of effective symmetry breaking, which can be readily applied to other inverse problems with symmetries.

2 Phase Retrieval (PR)

2.1 Symmetries in PR

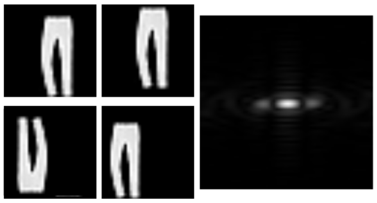

Figure 2: Symmetries in 2D PR

Phase retrieval (PR) is a central but decade-old unsolved problem in computational imaging with numerous applications (Shechtman et al., 2015). Here, we focus on the 2D version, which is probably most frequent in applications. Given $\boldsymbol{X} \in \mathbb{C}^{n_1 \times n_2}$, the forward mapping is

$$\boldsymbol{Y} = \left|\boldsymbol{F}_{m_1 \times n_1} \boldsymbol{X} \boldsymbol{F}_{m_2 \times n_2}^{\intercal}\right|^2 \in \mathbb{R}^{m_1 \times m_2}, \qquad (2.1)$$

where $\boldsymbol{F}_{m_1 \times n_1}$ consists of the first n_1 columns of the Fourier matrix $\boldsymbol{F}_{m_1 \times m_1}$, often called partial Fourier matrix; similarly for $\boldsymbol{F}_{m_2 \times n_2}$. The mapping is *generically* injective when $m_1 \geq 2n_1 - 1$ and $m_2 \geq 2n_2 - 1$, up to three intrinsic symmetries: 1) 2D translation

[1] Interestingly, the more train samples one gathers, the more serious the problem is.

of the nonzero content of $\boldsymbol{X}$; 2) conjugate of 2D flipping of $\boldsymbol{X}$; and 3) global phase transfer to $\boldsymbol{X}$: $\boldsymbol{X}e^{i\theta}$ for any $\theta \in [0, 2\pi)$. Any composition of these changes to $\boldsymbol{X}$ will leave the observation $\boldsymbol{Y}$ unaltered. Fig. 2 illustrate the first two symmetries, assuming $\boldsymbol{X}$ is a real-valued image.

It is easy to see that these symmetries can relate remote $\boldsymbol{X}$'s to nearby $\boldsymbol{Y}$'s. Applying our argument above, there will be learning difficulty due to the symmetries when deploying the end-to-end learning approach. Below, we will work with two simplified versions of PR first to illustrate the key ideas for symmetry breaking, and then discuss how to deal with the symmetries in the original PR.

Real Gaussian PR The forward model: $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$, where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is iid real Gaussian. The absolute-square operator $|\cdot|^2$ is applied elementwise. The only symmetry is sign, and $\boldsymbol{x}$ and $-\boldsymbol{x}$ are mapped to the same $\boldsymbol{y}$.

Complex Gaussian PR The forward model: $\boldsymbol{y} = |\boldsymbol{A}\boldsymbol{x}|^2$, where $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, and $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is iid complex Gaussian. The modulus-square operator $|\cdot|^2$ is applied elementwise. The only symmetry is global phase shift, and $e^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $\boldsymbol{y}$.

These two versions have been intensively studied in the recent developments of generalized PR; see, e.g., (Candes et al., 2015; Sun et al., 2017; Fannjiang & Strohmer, 2020).

2.2 Real Gaussian PR

In learning the square root example, there is a sign symmetry and we can break it by restricting all desired network outputs to be positive. Here, the symmetry is the global sign of vectors and antipodal points are mapped to the same observation. Thus, an intuitive generalization is breaking antipodal point pairs, and a simple solution is to make a hyperplane cut and take samples from only one side of the hyperplane! This is illustrated in Fig. 3 where we use the xy-hyperplane in $\mathbb{R}^3$.

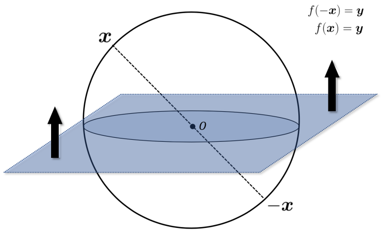
$f(-\boldsymbol{x}) = \boldsymbol{y}$
$f(\boldsymbol{x}) = \boldsymbol{y}$
$\boldsymbol{x}$
o
$-\boldsymbol{x}$

Figure 3: Symmetry breaking for real Gaussian PR.

In $\mathbb{R}^3$, the upper half space cut out by the xy-plane is connected. Moreover, it is representative as any point in the space (except for the plane itself) can be represented by a point in this set by appropriate global sign adjustment, and it cannot be made smaller to remain representative. The following proposition says that these properties also hold for high-dimensional spaces.

Proposition 2.1. *Let $\mathscr{R} \doteq \{\boldsymbol{x} \in \mathbb{R}^n : x_n > 0\}$, $Z \doteq \{\boldsymbol{x} \in \mathbb{R}^n : x_n = 0\}$. The following properties hold: 1)* ***(connected)*** *$\mathscr{R}$ is connected in $\mathbb{R}^n$; 2)* ***(representative)*** *Z is of measure zero (Rudin, 2006) and for any $\boldsymbol{x} \in \mathbb{R}^n \setminus Z$, either $\boldsymbol{x} \in \mathscr{R}$ or $-\boldsymbol{x} \in \mathscr{R}$; and 3)* ***(smallest)*** *No $\boldsymbol{x} \in \mathscr{R}$ can be represented by points in $\mathscr{R} \setminus \{\boldsymbol{x}\}$.*

Proof. See Appendix A.1. □

The coordinate hyperplane Z we use is arbitrary, and we can prove similar results for arbitrary hyperplanes. The set Z is negligible (i.e., has zero measure), and so the probability of sampling a point exactly from Z is zero. In fact, we can break the symmetry in Z also by recursively applying the current idea. For the sake of simplicity, we will not pursue it here.

Now we can apply the above result to preprocess the training samples $\{\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2\}$ for symmetry breaking: for all $\boldsymbol{x}_i$'s, if $\boldsymbol{x}_i$ lies above Z, we simply leave it untouched; if $\boldsymbol{x}_i$ lies below Z, we switch the sign of $\boldsymbol{x}_i$; if $\boldsymbol{x}_i$ happens to lie on Z, we make a small perturbation to $\boldsymbol{x}_i$ and then adjust the sign as before. Now $\boldsymbol{x}_i \in \mathscr{R}$ for all i. Since $\mathscr{R}$ is a connected set, when there are sufficiently dense training samples, small perturbations to $|\boldsymbol{A}\boldsymbol{x}_i|^2$ always lead to only small perturbations to $\boldsymbol{x}_i$. So we now have a nicely behaved underlying function.

The three properties are also necessary for effective symmetry breaking. Being representative is easy to understand. If the representative set is not the smallest, symmetry remains for certain points in the set and so symmetry breaking is not complete. Now the set can be smallest representative but not connected. An example in the setting of Theorem 2.1 would be taking out a $B \subsetneq \mathscr{R}$, and considering $M \doteq (-B) \cup (\mathscr{R} \setminus B)$. It is easy to verify that M is smallest representative, but not connected. This leaves us the trouble of approximating (locally) oscillatory functions.

2.3 Complex Gaussian PR

We now move to the complex case and deal with a different kind of symmetry. Recall that in the complex Gaussian PR, $e^{i\theta}\boldsymbol{x}$ for all $\theta \in [0, 2\pi)$ are mapped to the same $|\boldsymbol{A}\boldsymbol{x}|^2$, i.e., global phase shift is the symmetry. These "equivalent" points form a continuous curve in the complex space, contrasting the antipodal point pairs in the real case. Inspired by the real version, we seek the three properties in symmetry breaking.

To construct a smallest representative set for $\mathbb{C}^n$, it is helpful to start with low dimensions. When $n = 1$, any ray stemming from the origin (with origin removed) is a smallest representative subset for $\mathbb{C}$. For simplicity, we can take the positive axis $\mathbb{R}_+$. When $n = 2$, it is natural to use the building block $\mathbb{R}_+$ for $\mathbb{C}$ and start to consider product constructions of the form $\mathbb{R}_+ \times T \subset \mathbb{C}^2$ with $T \in \mathbb{C}$. Similarly for high dimensions, we try constructions of the form $\mathbb{R}_+ \times T \subset \mathbb{C}^n$ with $T \in \mathbb{C}^{n-1}$. Another consideration is the measure-zero set. In the real case, we used a coordinate hyperplane. Here, as a natural generalization, we take a complex hyperplane:

$$Z = \{\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{C}^n : x_1 = 0\}. \tag{2.2}$$

The question now is how to choose T to make $\mathbb{R}_+ \times T$ a smallest representative subset for $\mathbb{C}^n \setminus Z$. It turns out we actually do not get many choices. The following result says that real positivity assumed for the first coordinate constrains the construction significantly and the rest of coordinates are forced to be the entire complex space $\mathbb{C}^{n-1}$.

Proposition 2.2. *If $S \doteq \mathbb{R}_+ \times T$ with $T \subset \mathbb{C}^{n-1}$ is a representative subset for $\mathbb{C}^n \setminus Z$, then $T = \mathbb{C}^{n-1}$.*

Proof. See Appendix A.2. □

So we have the following construction.

Proposition 2.3. *The set $\mathscr{R} \doteq \{\boldsymbol{x} \in \mathbb{C}^n : \mathrm{Im}(x_1) = 0, x_1 > 0\}$ is a connected, smallest representative set for $\mathbb{C}^n \setminus Z$ with $Z \doteq \{\boldsymbol{x} \in \mathbb{C}^n : x_1 = 0\}$. Moreover, Z is a measure-zero subset of $\mathbb{C}^n$.*

Proof. See Appendix A.3. □

Here $\mathscr{R}$ is restricted half-space and enjoys the three desired properties, similar to the real case, despite the different symmetries. If we emulate the argument for the real case and preprocess the training data, the construction leads to effective symmetry breaking for complex Gaussian PR.

2.4 Fourier PR

Now we apply our insight gathered so far, especially the principle of constructing connected smallest representative set for symmetry breaking, to the original PR, which has three symmetries as discussed in Section 2.1.

Under the global phase transfer, equivalent data points form continuous curves as we discussed for complex Gaussian PR. These are relatively easy to represent algebraically. The conjugate 2D flipping and nonzero content translation, however, induce irregular equivalent sets that are shaped by the numbers inside $\boldsymbol{X}$ and hard to represent. Prescribing a rule for symmetry breaking in the original $\boldsymbol{X}$ space seems hopeless.

Fortunately, the three symmetries can be equivalently represented in terms of the complex phase $e^{i\theta}$ after the Fourier transform. Let $\mathcal{X}$ denote the oversampled Fourier transform of $\boldsymbol{X}$. Now 1) for 2D translation, any allowable[2] 2D translation $t_1, t_2 \in \mathbb{Z}$ induces the change $\mathcal{X}(k_1, k_2) \mapsto$

[2] The nonzero content cannot translate outside the boundaries.

$e^{i2\pi\left(\frac{k_1 t_1}{m_1}+\frac{k_2 t_2}{m_2}\right)}\mathcal{X}(k_1, k_2)$; 2) conjugate 2D flipping induces the change $\mathcal{X} \mapsto \overline{\mathcal{X}}$, i.e., change to the complex phase $e^{i\boldsymbol{\Theta}} \mapsto e^{-i\boldsymbol{\Theta}}$; and 3) global phase transfer induces the change $\mathcal{X} \mapsto e^{i\theta}\mathcal{X}$. The change due to 2) is a global sign flipping in the *angle* space (remember real Gaussian PR), and due to 3) is a line in the *angle* space. The equivalent sets are easy to represent in the angle space, but we take an equivalent representation in the complex phase space to avoid the tricky issue of dealing with the 2π periodicity in the angle space. 1) is still tricky whether in the angle or complex phase space.

Our overall strategy will be a combination of rigorous symmetry breaking for 2) and 3) in the complex phase space and *heuristic* symmetry breaking for 1) in the original space—our later real-data experiments will show the combination is effective. To break 1), we propose to simply center the nonzero content as a heuristic. To break 2) and 3), we perform the geometric construction in the angle space and then translate it back to the phase space representation. For the sake of space, we omit intuition behind the construction and directly present the results as follows.

Consider the following set in the phase domain

$$\mathscr{R} \doteq \left\{\boldsymbol{\Omega} \in \mathbb{C}^{m_1\times m_2} \;:\; \Omega(1,1)=1,\; \Omega(1,2)\in\mathbb{S}_+, \Omega(i,j)\in\mathbb{S}\;\forall\text{ other index }(i,j)\right\}, \qquad (2.3)$$

where $\mathbb{S}$ denotes the 1D complex circle and $\mathbb{S}_+$ the upper half circle. Formally, $\mathscr{R}$ can be understood as a set

$$\begin{bmatrix} \{1\} & \mathbb{S}_+ & \mathbb{S} & \cdots & \mathbb{S} \\ \mathbb{S} & \mathbb{S} & \mathbb{S} & \cdots & \mathbb{S} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{S} & \mathbb{S} & \mathbb{S} & \cdots & \mathbb{S} \end{bmatrix}_{m_1\times m_2}. \qquad (2.4)$$

We can prove the following, stated in the equivalent vector spaces for convenience. We write $\mathcal{R} \in \mathbb{S}^{m_1\times m_2}$ to mean the equivalent of $\mathscr{R}$ in the vector notation.

Theorem 2.4. *Consider the conjugate flipping and global phase transfer symmetries only. The set $\mathcal{R}$ is a connected, smallest representative in the phase domain $\mathbb{S}^{M^2}$ with a negligible set $\mathcal{N} = \{1\}\times\{\omega\in\mathbb{S} : \mathrm{Im}(\omega)=0\}^{m_1\times m_2-1}$.*

Proof. See Appendix A.4. □

To apply this, we work with end-to-end DNNs that directly predicts the $m_1 \times m_2$ complex phases. We first center the nonzero content inside $\boldsymbol{X}_i$'s in the training set, and then take the oversampled Fourier transform and perform the symmetry breaking as implied by Theorem 2.4 in the complex phase space. For any phase matrix $\boldsymbol{\Omega}$, the symmetry breaking goes naturally as follows: first a global phase transfer is performed to make $\Omega(1,1)=1$, and then perform a global angle negation, i.e., $\theta \mapsto -\theta$ of the second angle is negative, i.e., here we assume the angle has been transferred to the range of $(-\pi, \pi]$.

2.5 Symmetry breaking for general inverse problems

For general inverse problems, although the symmetries might be very different than here and the sample spaces could also be more complicated, the three properties, which concern only the geometric and topological aspects of the space, can be generalized as a basic mathematical principle for effective symmetry breaking. Our symmetry-breaking solution for Fourier PR also suggests that for problems with multiple symmetries, one may need to look at a transformed space, or even mixture of spaces for different symmetries for efficient representation and symmetry breaking.

3 Numerical Experiments

In this section, we set up various numerical experiments to verify our claim that effective symmetry breaking facilitates efficient learning. We start with Gaussian PR on synthetic data, and then move on the Fourier PR on real data. Particularly, we show that symmetry breaking substantially improve Fourier PR performance over alternative methods.

Table 1: Summary of results in terms of test error for real Gaussian PR. These numbers needs to be scaled by 10^{-4}. Blue coloring indicate the best performance in each row.

n	**Sample**	**NN-A**	K**-NN**	**NN-B**	**WNN-A**	K**-NN**	**WNN-B**	**DNN-A**	K**-NN**	**DNN-B**
5	2e4	10	17	283	**8**	18	283	10	19	284
	5e4	**6**	12	282	8	17	284	7	14	285
	1e5	**5**	10	284	**5**	12	283	13	18	284
	1e6	**4**	7	283	5	6	283	7	8	283
10	2e4	11	20	82	9	22	82	**8**	21	82
	5e4	9	16	82	**6**	18	82	9	20	82
	1e5	9	16	82	**6**	15	82	8	17	82
	1e6	7	13	82	**5**	10	82	9	11	82
15	2e4	12	17	38	**9**	16	38	**9**	16	38
	5e4	11	14	38	9	14	38	**8**	15	38
	1e5	10	13	38	8	13	38	**7**	13	38
	1e6	8	9	38	**7**	10	38	9	10	38

3.1 Real Gaussian PR

We take $m = 4n$ and draw iid uniformly random data points $\boldsymbol{x}_i$'s from the unit ball and consequently generate $\{(\boldsymbol{x}_i, |\boldsymbol{A}\boldsymbol{x}_i|^2)\}$ as the simulated datasets. All the datasets are split into 80% training and 20% test, and 10% of the training data are used for validation. We also vary the input dimension $n = 5, 10, 15$ and dataset size $2e4$, $5e4$, $1e5$, $1e6$. Since this is mostly for proof of concept, we do not go to higher dimensions. For all neural network models, we train them based on two variants of the training samples: one with symmetry untouched (i.e., before symmetry breaking) and one with symmetry breaking (i.e., after symmetry breaking). The former just leaves the samples unchanged, whereas the latter pre-processes the training samples using the procedures we described in Section 2 for the real Gaussian PR, respectively. To distinguish the two variants, we append our neural network model names with "-A" to indicate *after* symmetry breaking and "-B" to indicate *before* symmetry breaking. WNN and DNN are the wider and deeper versions of the basic neural network (NN) we use. For comparison, we also implement a baseline K nearest neighbor regression model. Model details, training parameters, and detailed evaluation metric can be found in Appendix A.6.

From the results summarized in Table 1, we observe that for the same NN architecture with any dimension-sample combination, symmetry breaking always leads to substantially improved performance. Without symmetry breaking, i.e., as shown in the $(\cdot)$-B columns, the estimation errors are always worse, if not significantly so, than the simple baseline K-NN model. By contrast, symmetry breaking as shown in the $(\cdot)$-A columns always leads to improved performance compared to the baseline. We observe similar patterns on complex Gaussian PR also, as detailed in Appendix A.6.

3.2 Fourier PR

We conduct our real-data experiments first on the Fashion-MNIST dataset (Xiao et al., 2017). We take their $60,000$ training images and $10,000$ test images to construct our training and test sets respectively. Each example is a 28×28 grayscale image. To simulate the typical black ground that causes the translation freedom in PR applications, we place all the images in a black background of $42 * 42$—most previous methods overlook this in their experiments, and practically the translation freedom, or what PR community call support estimation, is a major failure factor for most PR methods. So $n = 42$, and we take $m = 96$ here to ensure injectivity of the forward model $2n - 1 = 83$ is exceeded. We create 4 variants of the dataset to test the impact of symmetries on learning—this is the first time this kind of rigorous evaluation is performed; previous methods predominately use natural image datasets without translation freedom or with natural orientation, which does not match the scenarios in PR applications e.g., in coherent diffraction imaging. We do this by modifying the images as described below, followed by the standard operation of taking Fourier magnitudes.

- **No Symmetry**: i.e., all images are placed in the center of the black background i.e. padding of 7 on all side; samples shown in Fig. 4 (a)-left;

- **Flipping symmetry**: all images are placed in the center of the black background and 50% of randomly selected training and test images are top-down and left-right flipped; samples shown in Fig. 4 (b)-left.
- **Shift symmetry**: all images placed in a larger dark background and randomly translated; samples shown in Fig. 4 (c)-left;
- **Shift and Flipping symmetries**: random flipping followed by random translation; samples shown in Fig. 4 (d)-left.

Results on randomly selected test images are presented in Fig. 4. For each variant of the dataset, the left column is the groundtruth image, and the middle and right columns are results produced by U-Net-B (i.e., without symmetry breaking; this is exactly the method used in (Sinha et al., 2017)) and U-Net-A (i.e., with symmetry breaking—our method), respectively.

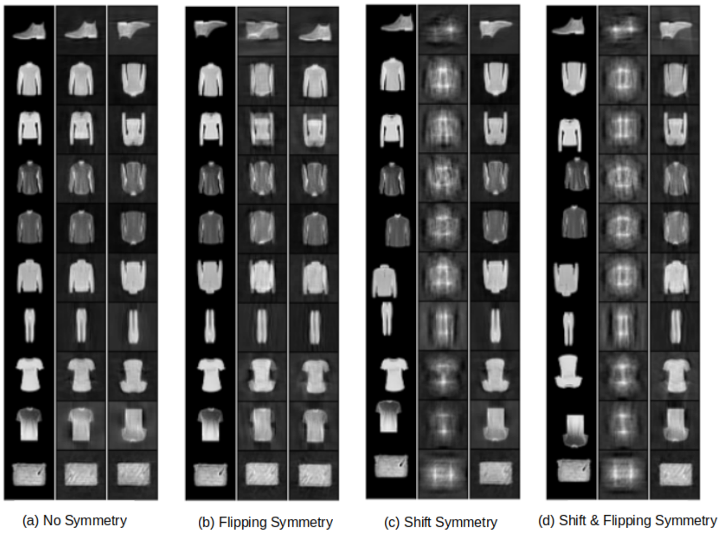

Figure 4: Visualization of recovery results. For each group, the first columns contain the groundtruth images. Second and third columns are reconstructions produced by U-Net-B and U-Net-A, respectively.

First note that with no symmetry, U-Net-B as a representative end-to-end method for PR gives good recovery, but it fails once the dataset contains the essential symmetries. The mode of failure is interesting, as the estimated images are almost always the superposition of the symmetric (translated or flipped) copies of the groundtruth. This is very similar to the failure model of the classic methods on PR. Moreover, for images that are visually similar between the original and the flipped copy e.g: "handbag", "leggings", the reconstruction results are good with or without the flipping symmetry, consistent with our intuition. Table 2 provides the average MSE (defined in Appendix A.6).

Table 2: Test error on different variants of fashion-mnist dataset

Method	U-Net-B	U-Net-A
No Symmetry	0.103	0.103
Flipping Symmetry	0.168	0.162
Shift Symmetry	0.249	0.102
Shift & Flipping Symmetry	0.248	0.161

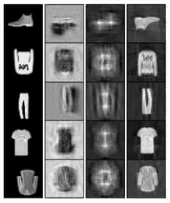

Figure 5: Visualization Comparison with ALM

4 Related Work

Table 3: Test error comparison with iterative method for randomly sampled 1000 fashion mnist test images containing shift flipping symmetry

Method	MSE
ALM	0.299
U-Net-B	0.249
U-Net-A	0.160

As alluded to above, recently there have been intensive research efforts on solving inverse problems using deep learning (McCann et al., 2017; Lucas et al., 2018; Arridge et al., 2019). The end-to-end approach is attractive not only because of its simplicity, but also because (i) we do not even need to know the forward models, so long as we can gather sufficiently many data samples and weak system properties such as symmetries—e.g., this is handy for complex imaging systems (Horisaki et al., 2016; Li et al., 2018); (ii) or alternatives have rarely worked, and a good example is Fourier PR (Fienup, 1982; Sinha et al., 2017).

Besides the linear inverse problems, the end-to-end deep learning approach has been empirically applied to a number of problems with symmetries, e.g., blind image deblurring (i.e., blind deconvolution) (Tao et al., 2018), real-valued Fourier phase retrieval (Sinha et al., 2017), 3D surface tangents and normal prediction (Huang et al., 2019), nonrigid structure-from-motion (Kong & Lucey, 2019; Wang et al., 2020). We believe that our work is the first to delineate the symmetry problem confronting effective learning and propose a solution principle that likely generalizes to other inverse problems.

For phase retrieval, the regularized optimization-deep learning hybrid approach has been applied by Metzler et al. (2018); Işıl et al. (2019), where HIO (a classic numerical method for PR) is still needed to produce good initialization and their methods mostly only perform local refinement—for simpler inverse problems, such special initialization is not required Ongie et al. (2020a). The end-to-end approach has also been applied by Goy et al. (2018); Uelwer et al. (2019); Metzler et al. (2020) with initial positive results. But as we discussed in the experiments, they do not seem to handle the essential difficulty caused by symmetries.

Mathematically, points related by symmetries form an equivalence class and these equivalence classes form a partition of the input space for the forward model. Our symmetric breaking task effectively consists in finding a *consistent* representation for the equivalence classes, where the consistency here requires the set of the representatives to be topologically connected.

REFERENCES

Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, may 2019. doi: 10.1017/s0962492919000059.

Tamir Bendory, Robert Beinert, and Yonina C. Eldar. Fourier phase retrieval: Uniqueness and algorithms. In *Compressed Sensing and its Applications*, pp. 55–91. Springer International Publishing, 2017. doi: 10.1007/978-3-319-69802-1_2.

Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pp. 437–478. Springer, 2012.

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, apr 2015. doi: 10.1109/tit.2015.2399924.

David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer New York, 2013. doi: 10.1007/978-1-4614-4942-3.

Pierre Comon. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. ACADEMIC PR INC, 2010. ISBN 0123747260.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pp. 184–199. Springer, 2014.

Dara Entekhabi, Hajime Nakamura, and Eni G Njoku. Solving the inverse problem for soil moisture and temperature profiles by sequential assimilation of multifrequency remotely sensed observations. *IEEE Transactions on Geoscience and Remote Sensing*, 32(2):438–448, 1994.

Albert Fannjiang and Thomas Strohmer. The numerics of phase retrieval. *arXiv:2004.05788*, 2020.

J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758, aug 1982. doi: 10.1364/ao.21.002758.

Rong Ge. *Provable Algorithms for Machine Learning Problems*. PhD thesis, Princeton University, 2013.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (4th Edition)*. Pearson, 2017. ISBN 978-0-13-335672-4.

Alexandre Goy, Kwabena Arthur, Shuai Li, and George Barbastathis. Low photon count phase retrieval using deep learning. *Physical Review Letters*, 121(24), dec 2018. doi: 10.1103/physrevlett.121.243902.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Gabor T. Herman. *Fundamentals of Computerized Tomography*. Springer London, 2009. doi: 10.1007/978-1-84628-723-7.

Ryoichi Horisaki, Ryosuke Takagi, and Jun Tanida. Learning-based imaging through scattering media. *Optics Express*, 24(13):13738, jun 2016. doi: 10.1364/oe.24.013738.

Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8638–8647, 2019.

Çağatay Işıl, Figen S. Oktem, and Aykut Koç. Deep iterative reconstruction for phase retrieval. *Applied Optics*, 58(20):5422, jul 2019. doi: 10.1364/ao.58.005422.

Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

John L Kelley. *General topology*. Courier Dover Publications, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andreas Kirsch. *An Introduction to the Mathematical Theory of Inverse Problems*. Springer New York, 2011. doi: 10.1007/978-1-4419-8474-6.

Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1558–1567, 2019.

Edmund Y. Lam and Joseph W. Goodman. Iterative statistical approach to blind image deconvolution. *Journal of the Optical Society of America A*, 17(7):1177, jul 2000. doi: 10.1364/josaa.17.001177.

Yunzhe Li, Yujia Xue, and Lei Tian. Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media. *Optica*, 5(10):1181, sep 2018. doi: 10.1364/optica.5.001181.

Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K. Katsaggelos. Using deep neural networks for inverse problems in imaging: Beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, jan 2018. doi: 10.1109/msp.2017.2760358.

Michael T. McCann, Kyong Hwan Jin, and Michael Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, nov 2017. doi: 10.1109/msp.2017.2739299.

Christopher A Metzler, Philip Schniter, Ashok Veeraraghavan, and Richard G Baraniuk. prdeep: Robust phase retrieval with a flexible deep network. *arXiv preprint arXiv:1803.00212*, 2018.

Christopher A. Metzler, Felix Heide, Prasana Rangarajan, Muralidhar Madabhushi Balaji, Aparna Viswanath, Ashok Veeraraghavan, and Richard G. Baraniuk. Deep-inverse correlography: towards real-time high-resolution non-line-of-sight imaging. *Optica*, 7(1):63, jan 2020. doi: 10.1364/optica.374026.

Ali Mousavi and Richard G Baraniuk. Learning to invert: Signal recovery via deep convolutional networks. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2272–2276. IEEE, 2017.

Gregory Ongie, Ajil Jalal, Christopher A Metzler Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 2020a.

Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, may 2020b. doi: 10.1109/jsait.2020.2991563.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, mar 2017. doi: 10.1007/s11633-017-1054-2.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Walter Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine*, 32(3):87–109, 2015.

Ayan Sinha, Justin Lee, Shuai Li, and George Barbastathis. Lensless computational imaging through deep learning. *Optica*, 4(9):1117, sep 2017. doi: 10.1364/optica.4.001117.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *Foundations of Computational Mathematics*, 18(5):1131–1198, aug 2017. doi: 10.1007/s10208-017-9365-9.

Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. doi: 10.1109/cvpr.2018.00853.

T. L. Tonellot and M. K. Broadhead. Sparse seismic deconvolution by method of orthogonal matching pursuit. In *72nd EAGE Conference and Exhibition incorporating SPE EUROPEC 2010*. EAGE Publications BV, jun 2010. doi: 10.3997/2214-4609.201401250.

Tobias Uelwer, Alexander Oberstraß, and Stefan Harmeling. Phase retrieval using conditional generative adversarial networks. *arXiv:1912.04981*, 2019.

Chaoyang Wang, Chen-Hsuan Lin, and Simon Lucey. Deep nrsfm++: Towards 3d reconstruction in the wild. *arXiv:2001.10090*, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pp. 341–349, 2012.

Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *Advances in neural information processing systems*, pp. 1790–1798, 2014.

A Deferred Proofs

A.1 Proof of Theorem 2.1

Proof. First recall the property that if any two points in a given set can be connected by a continuous path lying entirely in the set, then this set must be a connected set Kelley (2017). Now any two points $\boldsymbol{x}, \boldsymbol{y} \in R$ can be connected by the line segment $\{\alpha x + (1-\alpha)\boldsymbol{y} : \alpha \in [0,1]\} \subset R$. Thus R is connected. Moreover, $Z = \mathbb{R}^{n-1} \times \{0\}$ has Lebesgue measure zero since

$$\mu(Z) = \int_{\mathbb{R}^n} \mathbb{1}_Z \, d\boldsymbol{x} = \int_{\mathbb{R}^{n-1}} \left(\int_{\{0\}} \mathbb{1}_Z \, dx_n \right) d\boldsymbol{x}_{-n} = 0. \tag{A.1}$$

Here $\mathbb{1}_Z$ is the indicator function on Z, and $\boldsymbol{x}_{-n} \in \mathbb{R}^{n-1}$ is the vector formed by the first $n-1$ coordinates of $\boldsymbol{x}$. We used Tonelli's theorem Rudin (2006) to obtain the second equality, and the fact $\int_{\{0\}} \mathbb{1}_Z \, dx_n = 0$ to obtain the third equality. The rest of (2) is straightforward. For (3), suppose that there is another point $\widetilde{\boldsymbol{x}} \in R \setminus \{\boldsymbol{x}\}$ which can represent $\boldsymbol{x}$ up to a global sign flipping. Since both $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ are in R, which means they need to have the same sign for the last component, it must be $\boldsymbol{x} = \tilde{\boldsymbol{x}}$. We get a contradiction. □

A.2 Proof of Theorem 2.2

Proof. We prove by contradiction. Suppose that there is a $\boldsymbol{x}' \in \mathbb{C}^{n-1}$ but $\boldsymbol{x}' \notin T$. Then for any $x_1 \in \mathbb{R}_+$, $\boldsymbol{x} = (x_1; \boldsymbol{x}') \notin \mathbb{R}_+ \times T = S$ and $\boldsymbol{x} \in \mathbb{C}^{n-1} \setminus Z$. Since S is representative, we can find a $\theta \in [0, 2\pi)$ and $\widetilde{\boldsymbol{x}} \in S$ so that

$$\mathrm{e}^{i\theta} \boldsymbol{x} = \widetilde{\boldsymbol{x}}. \tag{A.2}$$

Since S has the first coordinate to be positive real numbers, by looking at the first component of Eq. (A.2) we have

$$\begin{cases} x_1 \cos\theta > 0 \\ x_1 \sin\theta = 0 \end{cases}, \tag{A.3}$$

from where we deduce that $\theta = 0$ and so $\boldsymbol{x} = \widetilde{\boldsymbol{x}} \in S$. This contradicts our construction that $\boldsymbol{x} \notin S$. □

A.3 PROOF OF THEOREM 2.3

Proof. First, Z has measure zero due to the same reason as in Eq. (A.1). Next, it is clear that any two points $\boldsymbol{x}, \boldsymbol{y} \in R$ can be connected by the line segment $\{\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y} : \alpha \in [0,1]\} \subset R$, and so R is a connected set. To see R is representative, for any $\boldsymbol{x} = (r_1 e^{i\theta_1}, x_2, \dots, x_n) \in \mathbb{C}^n \setminus Z$ where $r_1 > 0$, one can choose $\theta = 2\pi - \theta_1$ so that $e^{i\theta}\boldsymbol{x} \in R$. To show it is also smallest, we use a similar argument to that in Theorem 2.2. Let $\boldsymbol{x} \in R$ where we write $\boldsymbol{x} = (x_1; \boldsymbol{x}')$ with $\boldsymbol{x}' \in \mathbb{C}^{n-1}$. If another element $\widetilde{\boldsymbol{x}} \neq \boldsymbol{x} \in R$ can be represented by $\boldsymbol{x}$, namely, if there is $\theta \in [0, 2\pi)$ such that $\widetilde{\boldsymbol{x}} = \mathrm{e}^{i\theta}\boldsymbol{x}$, then we need to have $\mathrm{Im}(\mathrm{e}^{i\theta}x_1) = 0$ and $\mathrm{Re}(\mathrm{e}^{i\theta}x_1) > 0$. That is,

$$\begin{cases} x_1 \cos\theta > 0 \\ x_1 \sin\theta = 0 \end{cases} . \tag{A.4}$$

Since $x_1 > 0$, Eq. (A.4) implies that $\theta = 0$. But this contradicts with that $\boldsymbol{x} \neq \widetilde{\boldsymbol{x}}$ and thus no element in R can be represented by a distinct element in R. □

A.4 PROOF OF THEOREM 2.4

Proof. It is clear to see $\mathcal{R}$ is connected in $\mathbb{S}^{m_1 \times m_2}$ since $\mathscr{R}$ is path-connected set on $\mathbb{S}^{m_1 \times m_2}$ with the inherited subspace Euclidean topology of $\mathbb{C}^{m_1 \times m_2}$. Also, $\mathcal{N}$ is of Lebesgue measure 0 since it is a product of finite points. Now we are going to prove $\mathcal{R}$ is a representative of $\mathbb{S}^{m_1 \times m_2}$.

For any given $\boldsymbol{z} = (\mathrm{e}^{i\theta_0}, \mathrm{e}^{i\theta_1}, \cdots, \mathrm{e}^{i\theta_{m_1 \times m_2 - 1}})$, we need to find a $\boldsymbol{\omega} \in \mathcal{R}$ such that there is a $g \in \mathfrak{G}$ satisfying $g(\boldsymbol{\omega}) = \boldsymbol{z}$. If $\mathrm{Im}(\mathrm{e}^{i(\theta_1 - \theta_0)}) > 0$, we take $\boldsymbol{\omega} = (1, \mathrm{e}^{i(\theta_1 - \theta_0)}, \mathrm{e}^{i(\theta_2 - \theta_0)}, \cdots, \mathrm{e}^{i(\theta_{m_1 \times m_2 - 1} - \theta_0)})$ then $\boldsymbol{\omega} \in \mathcal{R}$ and $\mathrm{e}^{i\theta_0}\boldsymbol{\omega} = \boldsymbol{z}$. On the other hand, if $\mathrm{Im}(\mathrm{e}^{i(\theta_1 - \theta_0)}) < 0$, we can consider the conjugate format $\boldsymbol{\omega} = (1, \overline{\mathrm{e}^{i(\theta_1 - \theta_0)}}, \overline{\mathrm{e}^{i(\theta_2 - \theta_0)}}, \cdots, \overline{\mathrm{e}^{i(\theta_{m_1 \times m_2 - 1} - \theta_0)}})$ and it still possesses the same properties as the former case. This proves that is is representative.

At last, we need to show the smallestness in the sense that with any point of $\mathcal{R}$ removed, we cannot recover it by other points in $\mathcal{R}$. That is, with arbitrary $\tilde{\boldsymbol{z}} \in \mathcal{R}$ given, for any $g \in \mathfrak{G}$ and $\boldsymbol{z} \in \mathcal{R} \setminus \{\tilde{\boldsymbol{z}}\}$, we have $g(\boldsymbol{z}) \neq \tilde{\boldsymbol{z}}$. Before going to the details of the proof, we formulate an auxiliary lemma which helps simply the messy operations set $\mathfrak{G}$.

Lemma A.1. *Any operation in $\mathfrak{G}$ can be reformulated into a sequence of operations with the first part contains only subsequent phase transition operations and the second part contains only conjugate flipping operations (or reverse the order of these two parts). That is, for any $g \in \mathfrak{G}$, we can find $\tilde{g} \in \mathfrak{G}$ such that $\tilde{g} = \Phi \circ \Pi$ and $g = \tilde{g}$, where Π represents phase transition with total angles Π and Φ represents the flipping either odd times or even times (i.e., the identity).*

The proof of this lemma is straight forward and is demonstrated as following. First we may consider the simplest cases: $g_1 = \psi_1 \circ f$, where π_1 is the phase transition by angle ψ_1 and f is the flipping. For any given θ, we want to find a $g_2 = f \circ \psi_2$ such that $g_2(\mathrm{e}^{i\theta}) = g_1(\mathrm{e}^{i\theta})$. The g_2 can be found by solving equation

$$-(\psi_2 + \theta) = -\theta + \psi_1 + 2\pi k$$

to get $\psi_2 = -\psi_1 - 2\pi k$ for some $k \in \mathbb{Z}$. For more than two composition of operations, we can reduce it iteratively.

Now we can go back to the proof of smallestness. Write $\tilde{\boldsymbol{z}} = (\mathrm{e}^{i\tilde{\theta_0}}, \mathrm{e}^{i\tilde{\theta_1}}, \cdots, \mathrm{e}^{i\tilde{\theta}_{m_1 \times m_2 - 1}})$ and $\boldsymbol{z} = (\mathrm{e}^{i\theta_0}, \mathrm{e}^{i\theta_1}, \cdots, \mathrm{e}^{i\theta_{m_1 \times m_2 - 1}})$ where $\tilde{\theta_0} = \theta_0 = 0$ and $\mathrm{Im}(\mathrm{e}^{i\tilde{\theta_1}})$, $\mathrm{Im}(\mathrm{e}^{i\theta_1}) > 0$. Suppose that there is a $g \in \mathfrak{G}$ such that $\tilde{\boldsymbol{z}} = g(\boldsymbol{z})$. By Lemma A.1, we may assume $g = f \circ \psi$ or $g = \psi$ where ψ is a phase transition with the total angles ψ and f is the conjugate flipping. If $g = f \circ \psi$, $\tilde{\boldsymbol{z}} = g(\boldsymbol{z})$ implies for any $0 \leq j \leq M - 1$

$$\tilde{\theta_j} = -(\psi + \theta_j) + 2\pi k_j, \tag{A.5}$$

for some $k_j \in \mathbb{Z}$. We can solve $\psi = 2\pi k_0$ as $j = 0$ and this implies $\tilde{\theta_j} = 2\pi(k_j - k_0) - \theta_j$ for $1 \leq j \leq M - 1$, especially, $\tilde{\theta_1} = 2\pi(k_1 - k_0) - \theta_1$. This contradict to the fact that $\mathrm{Im}(\mathrm{e}^{i\tilde{\theta_1}})$, $\mathrm{Im}(\mathrm{e}^{i\theta_1}) > 0$. If $g = \psi$, we then have the relationship

$$\tilde{\theta_j} = (\psi + \theta_j) + 2\pi k_j. \tag{A.6}$$

Again, we can solve $\psi = 2\pi k_0$ as $j = 0$ and this indicates that $\tilde{\boldsymbol{z}} = \boldsymbol{z}$ which contradicts the assumption. Hence, we prove the smallestness. □

A.5 Difficulty with symmetries: what happened?

In this section, we investigate several aspects of the neural networks in the hope that some aspect can potentially help overcome the learning difficulties with symmetries. Based on the above discussion, we focus on the NN-A model. Typically, besides the network size, performance of neural networks is also strongly affected by the mini-batch size, learning rate, and regularization. To analyze the impact of the latter three, we vary each one of the parameters while keeping the others fixed. We work with real PR only and expect the situation for complex PR to be similar. To keep a reasonably fast run time while not hurting the performance, we take $2e4$ data samples, which seems sufficient for the above results.

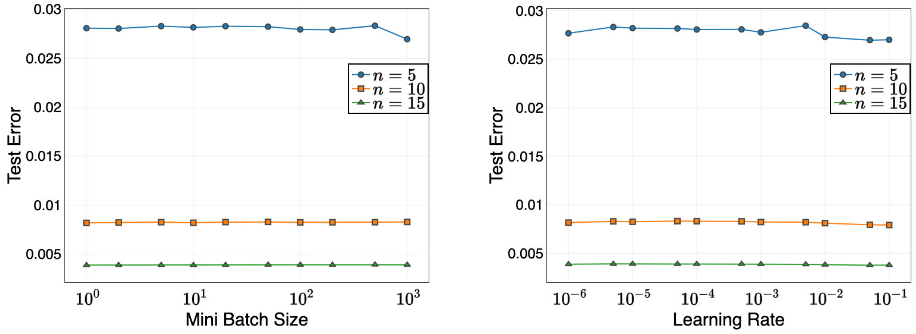

Figure 6: (Left) Test error vs. mini-batch size; (Right) Test error vs. learning rate.

Effect of mini-batch size The mini-batch size in stochastic optimization algorithms such as Adam that we use is considered to have a substantial impact on the performance of neural networks Bengio (2012). To see if this can help with the performance, we change the size to sweep several orders of magnitudes, i.e., $1e0, 1e1, 1e2, 1e3$, and also experiment with different dimensions, i.e., $n = 5, 10, 15$, on NN-B. From the results presented in Fig. 6 (Left), we conclude that varying the mini-batch size has a negligible effect on the test error.

Effect of learning rate The learning rate is the most critical hyper-parameter Jacobs (1988) that guides the change in model weights in response to the estimated error. To examine its effect on the test error, we vary it across six orders of magnitude: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and retrain NN-B. Again, the magnitude of the test error roughly remains the same across the distinct learning rates, as shown in Fig. 6 (Right).

Effect of regularization We explore three regularization schemes, $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_1 + \mathcal{L}_2$. Table 4 shows the results after our retraining of NN-B with the different schemes. It appears that no scheme clearly wins out.

Table 4: Test error using different regularization schemes

Regularization	$n = 5$	$n = 10$	$n = 15$
$\mathcal{L}_1$	0.02848	0.00831	0.00392
$\mathcal{L}_2$	0.02847	0.00830	0.00392
$\mathcal{L}_1 + \mathcal{L}_2$	0.02846	0.00830	0.00392

These results reinforce our claim that the bad performance of neural network learning without symmetry breaking is due to the intrinsic difficulty of approximating irregular functions, not due to suboptimal choice of neural network architecture or training hyper-parameters.

Table 5: Count of trainable parameters for $n = 15$

Models	Real	Complex
Neural Network	57,743	58,718
Wide Neural Network	197,391	199,326
Deep Neural Network	2,914,063	2,915,998

New MSE Due to the PR problem intrinsic symmetries, we'd like to correct the global phase before we evaluate the quality of the recovered images. Meanwhile, we need to scale the two images in the Error Function:

$$\min_{\boldsymbol{\theta},\boldsymbol{\eta}>0} \left\|\boldsymbol{A}-\boldsymbol{\eta}\boldsymbol{B}e^{i\boldsymbol{\theta}}\right\|_F^2 \tag{A.7}$$

where A is the original image in object domain and B is the recovered image in object domain. $\boldsymbol{eta}$ is scale varaible and $\boldsymbol{\theta}$ is a global phase.

Then we could expand the objective function based on complex-value rule:

$$\min_{\boldsymbol{\eta}>0} \ \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{\eta}\boldsymbol{B}\|_F^2 - 2\boldsymbol{\eta}Real(<\boldsymbol{A},\boldsymbol{B}e^{i\boldsymbol{\theta}}>) \tag{A.8}$$

which is equally to :

$$\min_{\boldsymbol{\eta}>0} \ \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{\eta}\boldsymbol{B}\|_F^2 - 2\boldsymbol{\eta}|<\boldsymbol{A},\boldsymbol{B}>| \tag{A.9}$$

First order optimal condition:

$$\nabla_{\boldsymbol{\eta}} \ \|\boldsymbol{A}\|_F^2 + \|\boldsymbol{\eta}\boldsymbol{B}\|_F^2 - 2\boldsymbol{\eta}|<\boldsymbol{A},\boldsymbol{B}>| = 0 \tag{A.10}$$

Deriving the objective function expansion form:

$$2\boldsymbol{\eta}\|B\|_F^2 - 2|<\boldsymbol{A},\boldsymbol{B}>| = 0 \tag{A.11}$$

Solution of η :

$$\boldsymbol{\eta} = \frac{|<\boldsymbol{A},\boldsymbol{B}>|}{\|B\|_F^2} \tag{A.12}$$

After substituting solution of $\boldsymbol{\eta}$, the solution of the error function is like:

$$\|\boldsymbol{A}\|_F^2 - \frac{|<\boldsymbol{A},\boldsymbol{B}>|}{\|B\|_F^2} \tag{A.13}$$

A.6 Gaussian-PR Experiments

Training and error metric The mean loss is used in the objective. We use the Adam optimizer (Kingma & Ba, 2014) and train all models for a maximum of 100 epochs. The learning rate is set as 0.001 by default and training is stopped if the validation loss does not reduce for 10 consecutive epochs. The validation set is also used for hyperparameter tuning. To train the models for the complex PR, real and complex parts of any complex vector are concatenated into a long real vector.

To imitate the real-world test scenario, we do not perform symmetry breaking on the test data. To measure performance, we use the normalized mean square error (MSE) which is rectified to account for the symmetry:

$$\varepsilon_{\text{real}} = \min_{s\in\{+1,-1\}} \frac{||\widehat{\boldsymbol{x}}s-\boldsymbol{x}||^2}{n}, \qquad \text{(real)} \tag{A.14}$$

$$\varepsilon_{\text{comp}} = \min_{\theta\in[0,2\pi)} \frac{||\widehat{\boldsymbol{x}}e^{i\theta}-\boldsymbol{x}||^2}{n}., \qquad \text{(complex)} \tag{A.15}$$

where $\widehat{\boldsymbol{x}}$ is the prediction by the learned models.

Results on randomly selected test images are presented in Fig. 4. For results on each variant of the dataset, the left column is the groundtruth image, and the middle and right columns are results produced by U-Net-B and U-Net-A, respectively. Table 2 provides test error. First note that with no symmetry, U-Net-B gives good recovery, but it fails on the variants containing symmetries. The mode of failure is interesting, as the estimated images are almost always the superposition of the symmetric copies of the groundtruth. This is very similar to the failure model of the classic methods on PR. Moreover, for images that are visually similar between the original and the flipped copy e.g: ”handbag”, ”leggings”, the reconstruction results are good with or without the flipping symmetry, consistent with our intuition.

On the other hand, irrespective of the symmetries, U-Net-A consistently leads to good recovery. Interestingly, U-Net-A can sometimes recover novel symmetric copies of the groundtruth image: e.g.,

Table 1 provides test errors for all models trained for real PR, and likewise Table 6 presents test errors for complex PR. All models for the same combination of input dimension n and sample size use the same set of data. Blues numbers in the tables indicate the best performing model across all the models in each row.

Learning models We set up an end-to-end pipeline and use neural network models to approximate the inverse mappings, as is typical done in this approach. The following are brief descriptions of the models used in our comparative study. Recall that in our problem setup, n is the dimension for $\boldsymbol{x}$ and m is the dimension for $\boldsymbol{y}$.

- **Neural Network (NN)**: fully connected feedforward NN with architecture m-256-128-64-n.
- **U-Net**: Fully Convolutional Network (Ronneberger et al., 2015), which is a state-of-the art model for image segmentation.
- **Wide Neural Network (WNN)**: we increase the size of hidden units of the NN by a factor of 2. The architecture is m-512-256-128-n.
- **Deep Neural Network (DNN)**: we increase the number and size of hidden layers of the NN by adding two more layers. The architecture is m-2048-1024-512-256-128-n.
- K**-Nearest Neighbors (**K**-NN)**: K-NN regression, where prediction is the average of the values of K nearest neighbors. In this work we use $K = 5$.

Table 6: Summary of results in terms of test error for complex Gaussian PR. These numbers needs to be scaled by 10^{-4}. Blue numbers indicate the best performance in each row.

n	Sample	NN-A	K-NN	NN-B	WNN-A	K-NN	WNN-B	DNN-A	K-NN	DNN-B
5	2e4	16	44	786	**11**	87	882	13	45	699
	5e4	**10**	39	718	12	38	669	19	39	697
	1e5	**06**	21	473	32	34	942	11	13	854
	1e6	**05**	06	642	64	72	453	14	15	731
10	2e4	79	237	452	65	80	453	**61**	239	380
	5e4	**56**	66	428	89	191	419	82	181	400
	1e5	97	139	436	**28**	58	431	55	86	453
	1e6	136	179	448	85	162	432	**77**	118	399
15	2e4	282	287	282	**129**	143	296	180	189	277
	5e4	192	272	313	**62**	126	308	172	233	313
	1e5	188	226	258	**141**	269	295	177	206	274
	1e6	136	179	448	**131**	184	395	182	202	283