

# Unsupervised Deep Unrolling Networks for Phase Unwrapping

Zhile Chen<sup>1,2</sup>

Yuhui Quan<sup>1,2\*</sup>

Hui Ji<sup>3</sup>

<sup>1</sup>School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

<sup>2</sup>Pazhou Lab, Guangzhou 510320, China

<sup>3</sup>Department of Mathematics, National University of Singapore, 119076, Singapore

cs\_zhilechen@mail.scut.edu.cn, csyhquan@scut.edu.cn, matjh@nus.edu.sg

## Abstract

*Phase unwrapping (PU) is a technique to reconstruct original phase images from their noisy wrapped counterparts, finding many applications in scientific imaging. Although supervised learning has shown promise in PU, its utility is limited in ground-truth (GT) scarce scenarios. This paper presents an unsupervised learning approach that eliminates the need for GTs during end-to-end training. Our approach leverages the insight that both the gradients and wrapped gradients of wrapped phases serve as noisy labels for GT phase gradients, along with sparse outliers induced by the wrapping operation. A recorruption-based self-reconstruction loss in the gradient domain is proposed to mitigate the adverse effects of label noise, complemented with a self-distillation loss for improved generalization. Additionally, by unfolding a variational model of PU that utilizes wrapped gradients of wrapped phases for its data-fitting term, we develop a deep unrolling network that encodes physics of phase wrapping and incorporates special treatments on outliers. In the experiments on three types of phase data, our approach outperforms existing GT-free methods and competes well against the supervised ones.*

## 1. Introduction

In many imaging systems, the direct acquisition of original phases of a target signal is both challenging and costly. Typically, initial measurement yields a phase image wrapped in  $[-\pi, \pi)$ . Let  $\mathbf{X} \in \mathbb{R}^{M \times N}$  and  $\mathbf{Y} \in [-\pi, \pi)^{M \times N}$  denote the true phase image and its noisy wrapped counterpart, respectively. The wrapping process can be formulated as:

$$\mathbf{Y} = \mathcal{W}(\mathbf{X} + \mathbf{N}), \quad \mathcal{W} : \mathbb{R}^{M \times N} \rightarrow [-\pi, \pi)^{M \times N}, \quad (1)$$

\*Corresponding author: Yuhui Quan.

This work is supported by National Natural Science Foundation of China (Grant No. 62372186), Natural Science Foundation of Guangdong Province (Grant No. 2022A1515011755, 2023A1515012841), Fundamental Research Funds for Central Universities (Grant No. x2jsD2230220), and Singapore MOE AcRF Tier 1 (Grant No. A-8000981-00-00).

where  $\mathcal{W}$  denotes the wrapping operator with its entry-wise operation defined as  $\mathcal{W}(\theta) = ((\theta + \pi) \bmod 2\pi) - \pi$  for  $\theta \in \mathbb{R}$ , and  $\mathbf{N} \in \mathbb{R}^{M \times N}$  denotes the measurement noise, typically assumed to be Gaussian [22]. Consequently, the original phase image, which exhibits strong smoothness, is degraded to a wrapped version with both incorrect values and artificial discontinuities induced by the modulo operation, which adversely affects subsequent processing.

Phase unwrapping (PU) is about recovering the original unwrapped phase image  $\mathbf{X}$  (up to an additive constant) from its noisy wrapped counterpart  $\mathbf{Y}$ . Serving as a critical step in many optical imaging techniques for generating clear and coherent phase images, PU finds extensive applications in diverse domains such as quantitative phase imaging, magnetic resonance imaging, synthetic aperture radar interferometry, 3D depth sensing, phase contrast microscopy, fringe projection, and digital holographic interferometry; see e.g. [10, 14, 16, 23, 41, 47, 56, 57].

The model (1) is also usually formulated as follows:

$$\mathbf{Y} = \mathbf{X} - 2\pi\mathbf{K} + \mathbf{N}', \quad \mathbf{K} \in \mathbb{Z}^{M \times N}, \quad (2)$$

where  $\mathbf{K} \in \mathbb{Z}^{M \times N}$  denotes the map of wrap counts, indicating the number of times a phase value has been wrapped around by  $2\pi$ , and  $\mathbf{N}'$  denotes the noise dependent on both  $\mathbf{X}$  and  $\mathbf{N}$ , whose distribution can be complex even when  $\mathbf{N}$  follows a simple distribution such as the normal distribution. It can be seen that PU is a challenging ill-posed inverse problem involving a continuous variable  $\mathbf{X}$  and a discrete variable  $\mathbf{K}$ . Direct solutions by integrating wrapped phase differences are insufficient for ensuring spatial consistency in all directions of the unwrapped phases and may accumulate significant errors due to substantial noise.

**Limitations of existing works:** Conventional approaches for PU include reliability-guided methods (e.g. [3, 9, 11, 23, 36]), filtering-based methods (e.g. [2, 45]), and model-driven methods (e.g. [1, 12, 15]). These approaches are either sensitive to measurement noise, due to their heuristic rules, or overly simplistic for complex phase structures, due to their handcrafted image priors. In contrast, deep learning

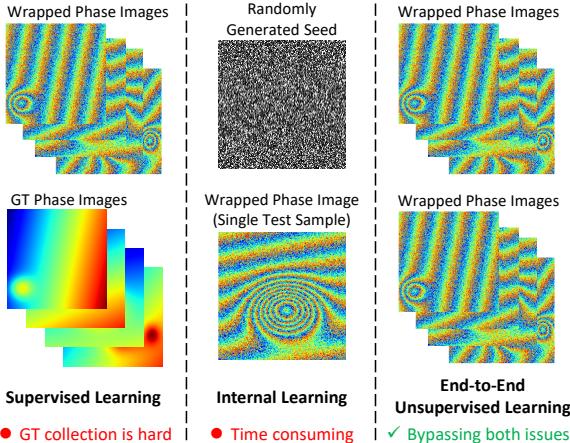


Figure 1. Training data setting of existing PU methods versus ours.

(DL) with neural networks (NNs) has emerged as a powerful tool for PU [41]. A pre-trained end-to-end NN offers data adaptivity, rapid inference, and excellent expressivity.

Existing DL-based PU methods often rely on *supervised learning* with ground-truth (GT) phases. However, acquiring GT phases is often impractical due to physical limits in imaging, destructive imaging techniques, and ethical/safety concerns. This challenge limits the applicability of supervised DL in many practical scenarios. *Internal unsupervised DL*-based methods (*e.g.* [47]) use an untrained NN to parameterize latent phases, adjusting it to fit observed measurements with certain regularization. Though flexible, this kind of methods requires time-costly per-sample test-time training, resulting in poor computational efficiency compared to supervised DL-based methods.

**Our approach:** To bypass the difficulty of GT acquisition in supervised DL and avoid the computational inefficiency of internal unsupervised DL, we present an *external unsupervised DL* approach for PU. It trains an end-to-end NN solely on a dataset of noisy wrapped measurements. See Fig. 1 for the comparison in training data configuration between existing works and ours. Indeed, our study is the first one on external (end-to-end) unsupervised DL for PU.

Our approach is inspired by the insight that applying the gradient operator or the wrapping gradient operator to a wrapped phase image will result in an output identical to the gradient map of the GT phase image, with the exception of a sparse outlier set. Concretely, we have that

- (a)  $\nabla Y[m, n] = \nabla X[m, n] + \nabla N[m, n]$  holds for the points whose neighbors share the same wrap count.
- (b)  $W(\nabla Y[m, n]) = \nabla X[m, n] + \nabla N[m, n]$  is applicable for the points that meet the 2D Itoh’s continuity condition [13].

The majority of points in a GT phase image fall under the aforementioned categories. Therefore, on these points,  $\nabla Y$  or  $W(\nabla Y)$  can serve as a noisy label of  $\nabla X$ . Once  $\nabla X$

is predicted, reconstructing  $X$  is relatively simple.

Our approach utilizes  $\nabla Y$  for unsupervised training and tackles its label noise via a recorruption-based self-reconstruction loss function in the gradient domain, which is inspired by existing recorruption-based denoisers [19, 21, 46]. In addition, to mitigate the potential ambiguity and overfitting caused by the absence of GTs, a self-distillation loss function is proposed that leverages intermediate predictions from the NN for pseudo-supervision.

To encode physical model/priors of phase wrapping into the NN-learned PU process, which is beneficial for mitigating possible overfitting particularly when training on a limited amount of data, we develop a deep unrolling network (DUN) by unfolding a variational model that leverages  $W(\nabla Y)$  for data fitting. To our knowledge, this is the first work to utilize deep unrolling for PU. Indeed, most existing studies rely on U-shaped convolutional NNs (CNNs) or recurrent NNs (RNNs) that ignore physical priors of phase wrapping. Additionally, our DUN is tailored to handle the outliers that do not satisfy the Itoh’s continuity condition, thereby mitigating their potential adverse effects.

**Main contributions:** Our contributions include

- We introduce the first external unsupervised DL approach for end-to-end PU, circumventing the challenge of GT acquisition in supervised DL and the computational inefficiency of internal unsupervised DL.
- We perform the first exploration of deep unfolding for PU, founded on a variational model that utilizes wrapped gradients of wrapped phases for data fitting under Itoh’s continuity condition. The resulting DUN is outlier-aware and enjoys advantages in interpretability and performance.
- We propose a recorruption-based self-reconstruction loss function with noise tolerance to leverage Itoh’s continuity condition, as well as a self-distillation loss function for improved generalization.

In extensive experiments, our proposed approach shows superior performance over existing GT-free PU methods and competes well against the supervised ones.

## 2. Related Work

**Non-learning-based PU:** Methods along this line fall into three types: reliability-guided, filtering-based, and model-driven. Reliability-guided methods [3, 9, 11, 23, 36, 54] use a manually-designed metric to quantify pixel-wise reliability for path selection or region expansion during phase correction. Due to unreliable quality scores in noisy regions, quality-guided methods are noise-sensitive. Filtering-based methods [2, 45] treat PU as a smoothing problem, leveraging non-linear filters for denoising and unwrapping. However, their limited-support filters fail to exploit global consistency constraints within a phase image. Model-driven methods [1, 8, 12, 15] construct and solve an optimization

model of PU regularized by some handcrafted priors. While achieving strong global consistency brought by the global optimization, these methods often fall short for complex phase patterns, due to the use of handcrafted priors.

**Supervised DL for PU:** Supervised DL-based PU methods differ mainly in NN architectures. Existing methods can be roughly classified as CNN-based or RNN-based.

Leveraging the effectiveness of CNNs in segmentation and classification, many CNN-based methods (*e.g.* [10, 34, 35, 49, 50, 52, 56]) cast PU as a wrap count prediction problem, using a pixel-wise classification loss with wrap counts as labels. For robustness to noise, pre-denoising or post-processing modules are often included; see *e.g.* [34, 35, 52]. However, these methods face scalability issues with wide-range phase data, where the candidate values for wrap counts are overwhelmed. Instead, some CNN-based methods [6, 25, 40, 57]. Typically, CNN-based methods are inefficient in exploiting global spatial dependencies. In contrast, the RNN-based methods [22, 32] excel at capturing long-range dependencies. Viewing a phase pixel as an element in a sequence, these methods predefine several paths across all phase pixels to form an RNN. However, due to computational costs, only a few paths can be utilized. This constraint on path selection results in suboptimal outcomes.

The NN designs of most existing methods overlook the physical priors of phase wrapping, which are beneficial for generalization, especially when training data is limited. Our proposed DUN incorporates these priors, enhancing interpretability, reducing overfitting, and boosting performance.

**Unsupervised DL for PU:** Despite significant advances in unsupervised DL for various inverse problems in imaging, *e.g.*, denoising [17, 21, 26], deblurring [5, 18, 27, 31], compressed sensing [28, 38, 42] and phase retrieval [4, 29], unsupervised DL for PU has been much less studied. One related work is Yang *et al.* [47], which utilizes the deep image prior [39] encoded by an untrained CNN to address the absence of GT data. Specifically, the phase image is reparameterized by a CNN, with its weights learned to fit the measurements. As shown in [39, 47], the CNN prefers smooth patterns and requires early stopping for smooth phase unwrapping, akin to conventional model-based methods but employing a CNN-based prior. A drawback is the need for per-sample retraining, which is computationally intensive. In contrast, our approach allows end-to-end unsupervised training on a dataset, eliminating test-time learning and significantly speeding up test.

**DUNs for imaging:** DUNs have been studied for many inverse problems in imaging, *e.g.*, denoising [20, 55], super-resolution [51], deblurring [20], compressed sensing [24, 53] and medical imaging [7]. However, the study on deep unrolling for PU is scant. Our study is the first to develop DUNs for PU, in addition to an unsupervised training scheme.

### 3. Methodology

#### 3.1. Basics and Keys

Consider estimating  $\mathbf{X}$  from its wrapped version  $\mathcal{W}(\mathbf{X}) = \mathbf{X} - 2\pi\mathbf{K}$ ,  $\mathbf{K} \in \mathbb{Z}^{M \times N}$ . Let  $\nabla: \mathbf{Z} \rightarrow (\nabla_x \mathbf{Z}; \nabla_y \mathbf{Z})$  denote the spatial gradient operator. By definition,

$$\nabla \mathcal{W}(\mathbf{X})[m, n] = \nabla \mathbf{X}[m, n] + 2\pi \nabla \mathbf{K}[m, n]. \quad (3)$$

For any location  $[m, n]$  whose adjacent pixels for gradient calculation have the same wrap count, we have

$$\nabla \mathcal{W}(\mathbf{X})[m, n] = \nabla \mathbf{X}[m, n]. \quad (4)$$

In other words, the wrap count can be nullified by the gradient operator on many points. For the points not meeting the condition, we have the following Proposition.

**Proposition 1.** *[Two-dimensional extension of Itoh’s continuity condition [13]] For any point  $[m, n]$  satisfying*

$$\|\nabla \mathbf{X}[m, n]\|_\infty = \max\{|\nabla_x \mathbf{X}[m, n]|, |\nabla_y \mathbf{X}[m, n]|\} < \pi, \quad (5)$$

*we have that  $\mathcal{W}(\nabla \mathcal{W}(\mathbf{X}))[m, n] = \nabla \mathbf{X}[m, n]$ .*

*Proof.* See supplemental material for the proof.

Proposition 1 shows that the wrapped gradients of a wrapped phase image are identical to the gradients of the GT, for the points satisfying the 2D Itoh’s continuity condition (5) (called Itoh’s condition for brevity in the following text). In general, GT phases exhibit gradual changes in most points, and their rapid changes only occur at a few points sparsely distributed over the image. Then, the outlier points not meeting Itoh’s condition are sparse.

Now consider a noisy wrapped image  $\mathbf{Y} = \mathcal{W}(\mathbf{X} + \mathbf{N})$ . We can directly extend Eq. (4) to the noisy case:

$$\nabla \mathbf{Y}[m, n] = \nabla \mathbf{X}[m, n] + \nabla \mathbf{N}[m, n], \quad (6)$$

for any location  $[m, n]$  in  $(\mathbf{X} + \mathbf{N})$  whose adjacent points share the same wrap count. Similarly, we have the following corollary of Proposition 1.

**Corollary 1.1.** *For points  $[m, n]$  satisfying  $\|\nabla \mathbf{X}[m, n] + \nabla \mathbf{N}[m, n]\|_\infty < \pi$ , we have*

$$\mathcal{W}(\nabla \mathbf{Y}[m, n]) = \mathcal{W}(\nabla \mathbf{X}[m, n]) + \nabla \mathbf{N}[m, n]. \quad (7)$$

It is clear that both  $\nabla \mathbf{Y}$  and  $\mathcal{W}(\nabla \mathbf{Y})$  can act as a noisy version for  $\nabla \mathbf{X}$ , except on some generally-sparse outlier points that do not satisfy the corresponding conditions. If we can mitigate those noise and outliers to predict  $\nabla \mathbf{X}$ , the remaining task, reconstructing  $\mathbf{X}$  from  $\nabla \mathbf{X}$ , is straightforward, as  $\mathbf{X}$  can be reconstructed by integrating its spatial gradients  $\nabla \mathbf{X}$  with appropriate boundary conditions. Indeed, the primary concern in PU revolves around relative phases rather than the absolute ones. The observations described by Eq. (7) and Eq. (6) underpin our DUN and unsupervised loss function, respectively.

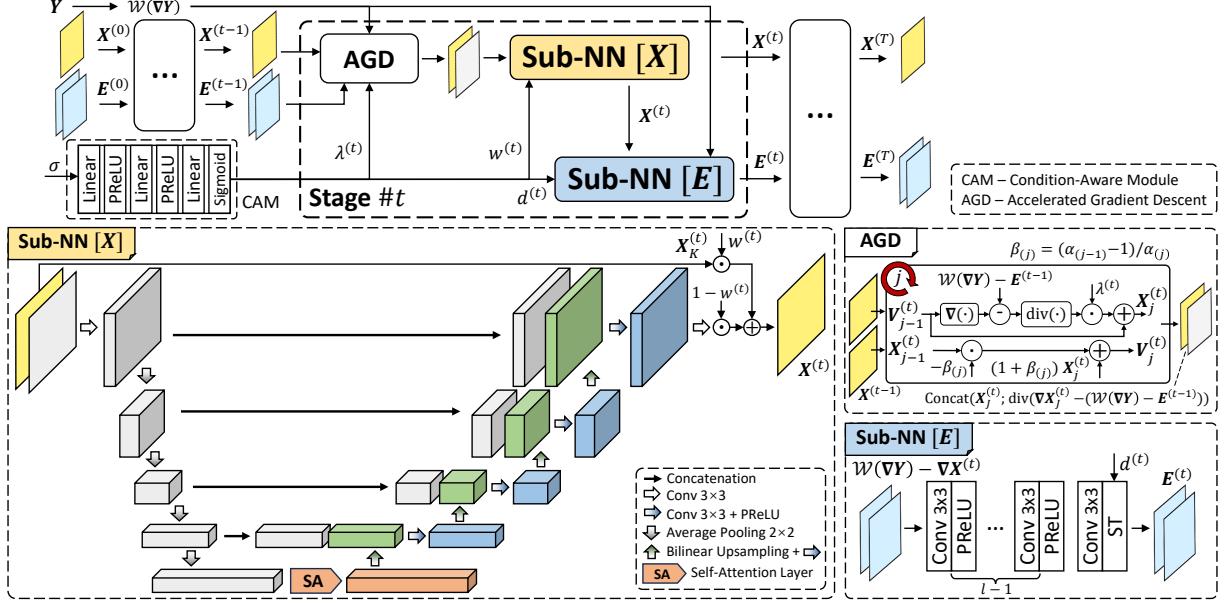


Figure 2. Architecture of our proposed DUN for PU.

### 3.2. Deep Unrolling Network for PU

Our DUN is based on a regularization model inspired by Eq. (7). Though Eq. (7) is inapplicable to the pixels not meeting Itoh's condition, these outliers are typically sparse in a phase image with gradual changes in most areas. Consequently, we introduce a sparse variable  $\mathbf{E} \in \mathbb{R}^{M \times N \times 2}$  into the regularization model, which counteracts the errors introduced by those sparse outliers. Specifically, we consider the following regularization model of PU:

$$\min_{\mathbf{X}, \mathbf{E}} \|\nabla \mathbf{X} - \mathcal{W}(\nabla \mathbf{Y}) + \mathbf{E}\|_F^2 + \phi(\mathbf{X}) + \psi(\mathbf{E}), \quad (8)$$

where  $\phi, \psi$  are some regularization functions for the latent phase image  $\mathbf{X}$  and the error matrix  $\mathbf{E}$ , respectively. It can be seen that  $\mathbf{E}$  essentially neutralizes outlier effects, preserving correctness of the data fitting term.

Our DUN is constructed via the unrolled iterative solver for (8), which reads as follows: for  $t = 1, \dots, T$ ,

$$\begin{cases} \mathbf{X}^{(t)} := \operatorname{argmin}_{\mathbf{X}} \|\nabla \mathbf{X} - (\mathcal{W}(\nabla \mathbf{Y}) - \mathbf{E}^{(t-1)})\|_F^2 + \phi(\mathbf{X}), \\ \mathbf{E}^{(t)} := \operatorname{argmin}_{\mathbf{E}} \|\mathbf{E} - (\mathcal{W}(\nabla \mathbf{Y}) - \nabla \mathbf{X}^{(t)})\|_F^2 + \psi(\mathbf{E}). \end{cases}$$

Accordingly, the DUN predicts  $\mathbf{X}$  from  $\mathcal{W}(\nabla \mathbf{Y})$ . It alternatively connects two types of sub-NNs that act as the estimators of  $\mathbf{X}^{(t)}$  and  $\mathbf{E}^{(t)}$ , respectively. See Fig. 2 for an illustration of this architecture.

**Sub-NN for estimating  $\mathbf{X}^{(t)}$ :** The sub-problem for  $\mathbf{X}^{(t)}$  is solved using proximal gradient descent (PGD), which combines gradient descent with proximal mapping. Considering the limited number of modules (i.e., iteration number  $T$ ) in our DUN for computational efficiency, we implement

multiple gradient descent steps within a stage to bridge the gap. Specifically, Nesterov's acceleration scheme is used to enhance the gradient descent, called AGD. We initialize with  $\mathbf{X}^{(0)} = \mathbf{X}^{(t-1)}, \mathbf{X}^{(0)} = \mathbf{1}, \mathbf{E}^{(0)} = \mathbf{0}$  and  $\alpha_{(0)} = 0$ . For  $j = 1, \dots, J$ ,

$$\begin{cases} \mathbf{X}_{(j)}^{(t)} = \mathbf{V}_{(j-1)}^{(t)} + \lambda^{(t)} \operatorname{div}(\nabla \mathbf{V}_{(j-1)}^{(t)} - (\mathcal{W}(\nabla \mathbf{Y}) - \mathbf{E}^{(t-1)})), \\ \alpha_{(j)} = \frac{1}{2}(1 + \sqrt{1 + 4\alpha_{(j-1)}^2}), \\ \mathbf{V}_{(j)}^{(t)} = \mathbf{X}_{(j)}^{(t)} + \frac{\alpha_{(j-1)} - 1}{\alpha_{(j)}}(\mathbf{X}_{(j)}^{(t)} - \mathbf{X}_{(j-1)}^{(t)}). \end{cases}$$

Here the term  $\operatorname{div}(\nabla \mathbf{V}_{(j-1)}^{(t)} - (\mathcal{W}(\nabla \mathbf{Y}) - \mathbf{E}^{(t-1)}))$  corresponds to the gradient  $\mathcal{G}_t(\mathbf{X}) = \nabla_{\mathbf{X}} \|\nabla \mathbf{X} - (\mathcal{W}(\nabla \mathbf{Y}) - \mathbf{E}^{(t-1)})\|_F^2$ . Afterward, the proximal operator, defined as  $\operatorname{Prox}_f(\bar{\mathbf{Z}}) := \operatorname{argmin}_{\mathbf{Z}} \|\mathbf{Z} - \bar{\mathbf{Z}}\|_2^2 + f(\mathbf{Z})$ , is applied:

$$\mathbf{X}^{(t)} = \operatorname{Prox}_{\phi}(\mathbf{X}_{(J)}^{(t)}). \quad (9)$$

This proximal operator is implemented using a sub-NN with an attentive U-shape structure. As illustrated in Fig. 2, it has a  $16 \times$  down-scaling encoder and a corresponding up-scaling decoder, connected by skip connections. At the coarsest scale, a standard self-attention block is used for exploiting global features. The sub-NN also takes the gradient term  $\mathcal{G}_t(\mathbf{X}_{(J)}^{(t)})$  as an additional input via concatenation. A long skip connection from the input to output is added, enhancing feature flow with a weighted summation.

**Sub-NN for estimating  $\mathbf{E}^{(t)}$ :** The sub-problem for  $\mathbf{E}^{(t)}$  aims to address the outliers presented in the data fitting term defined on  $\mathcal{W}(\nabla \mathbf{Y}) - \nabla \mathbf{X}^{(t)}$ , guided by the prior encoded in  $\psi(\mathbf{E})$ . We employ a sub-NN with  $\mathcal{W}(\nabla \mathbf{Y}) - \nabla \mathbf{X}^{(t)}$

as the input to implement the solver of this sub-problem. This sub-NN, illustrated in Fig. 2, is composed of sequential convolutional layers, each followed by a parametric rectified linear unit (PReLU). The sparsity prior of  $\mathbf{E}$  is induced by attaching a soft thresholding activation function, parametrized by a non-negative parameter  $\text{ReLU}(d^{(t)})$ .

**Condition-aware parameter estimation:** The noise level in the wrapped phases impacts the optimal selection of the parameters  $\{\lambda^{(t)}, w^{(t)}, d^{(t)}\}_{t=1}^T$ . Existing DUNs (e.g. [51]) often include imaging condition values in the input. In our DUN, a condition-aware module (CAM) is used to predict these parameters from noise level, using a chain of fully-connected layers with PReLU or Sigmoid activations.

### 3.3. Unsupervised End-to-End Training Scheme

Let  $\mathcal{F}$  denote our DUN that maps  $\mathcal{W}(\nabla \mathbf{Y})$  to  $\mathbf{X}$ . Our proposed loss function  $\mathcal{L}_{\text{total}}$  for the unsupervised training of  $\mathcal{F}$  comprises a noise-resistant self-reconstruction loss  $\mathcal{L}_{\text{sr}}$  and a self-distillation loss  $\mathcal{L}_{\text{sd}}$ , applied to each stage of the DUN:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sr}} + \eta \mathcal{L}_{\text{sd}}, \quad \eta \in \mathbb{R}^+. \quad (10)$$

**Noise-resistant self-reconstruction loss:** This loss utilizes  $\nabla \mathbf{Y}$  as a noisy label for  $\nabla \mathbf{X}$ , as shown in Eq. (6). To mitigate the impact of label noise, the loss is defined as

$$\mathcal{L} := \mathbb{E}_{\mathbf{U}} \|\nabla \mathcal{F}(\mathcal{W}(\nabla \mathbf{Y} + \nabla \mathbf{U})) - (\nabla \mathbf{Y} - \nabla \mathbf{U})\|_{\text{F}}^2, \quad (11)$$

where  $\mathbf{U}$  is random noise drawn from the distribution of  $\mathbf{N}$ . The introduction of the noise pair  $\{+\nabla \mathbf{U}, -\nabla \mathbf{U}\}$  is to achieve the resilience to measurement noise in  $\nabla \mathbf{Y}$  in the training loss, as justified in Proposition 2.

**Proposition 2.** Let  $\mathbf{Y} = \mathcal{W}(\mathbf{X} + \mathbf{N})$ . Suppose Eq. (6) is satisfied at all points. Assume that  $\mathbf{N}, \mathbf{U} \sim \mathcal{P}$  are independent. Then, we have that

$$\mathbb{E}_{\mathbf{Y}} \mathcal{L} = \mathbb{E}_{\mathbf{X}, \mathbf{N}, \mathbf{U}} \|\nabla \mathcal{F}(\mathcal{W}(\nabla \mathbf{Y} + \nabla \mathbf{U})) - \nabla \mathbf{X}\|_{\text{F}}^2 + C_0, \quad (12)$$

where  $C_0$  is a constant.

*Proof.* See the proof in supplemental material.

Proposition 2 shows that if one can independently draw instances  $\mathbf{U}$  from the same distribution as  $\mathbf{N}$ , the loss  $\mathcal{L}$  effectively simulates the loss supervised by  $\nabla \mathbf{X}$ , nullifying the label noise as shown in Eq. (12). Note that Proposition 2 requires all points satisfying Eq. (6), which is not generally true. On the outliers,  $\nabla \mathbf{Y}$  provides wrong supervision with superfluous multiples of  $2\pi$ , as seen in Eq. (3). To counteract the impact of those outlier points, we exploit the property  $\mathcal{W}(\theta + 2k\pi) = \theta$  to modify the self-reconstruction loss as follows:

$$\mathcal{L}_{\text{sr}} := \mathbb{E}_{\mathbf{U}} \|\mathcal{W}[\nabla \mathcal{F}(\mathcal{W}(\nabla \mathbf{Y} + \nabla \mathbf{U})) - (\nabla \mathbf{Y} - \nabla \mathbf{U})]\|_{\text{F}}^2. \quad (13)$$

That is, an extra  $\mathcal{W}$  is applied to the prediction residual within the Frobenius norm, which negates the superfluous multiples of  $2\pi$  during the loss calculation.

**Self-distillation loss:** Although  $\mathcal{L}_{\text{sr}}$  reduces the impact of measurement noise, the ambiguity in solutions due to wrapping may remain, particularly at those outlier points. Fortunately, CNNs including unrolling CNNs [30, 39], show strong induced bias towards regular structures, which aids in reducing such ambiguity during the training with  $\mathcal{L}_{\text{sr}}$ .

To further reduce the ambiguity, we introduce a self-distillation loss defined as

$$\mathcal{L}_{\text{sd}} := \mathbb{E}_{\mathbf{U}} \|\nabla \mathcal{F}(\mathcal{W}(\nabla \mathbf{Y})) - \nabla \bar{\mathcal{F}}(\mathcal{W}(\nabla \mathbf{Y} + \nabla \mathbf{U}))\|_{\text{F}}^2. \quad (14)$$

where  $\bar{\mathcal{F}}$  denotes the NN detached from the previous iteration with stop gradient. The loss leverages pseudo supervision from the NN's varied predictions with noise-altered inputs, implicitly aggregating those predictions for self-training. This self-distillation scheme can reduce the NN's prediction variance, enhancing the PU accuracy. Moreover, recall that  $\mathcal{L}_{\text{sr}}$  inputs  $\mathcal{W}(\nabla \mathbf{Y} + \nabla \mathbf{U})$  to  $\mathcal{F}$ , which does not fully align with  $\mathcal{W}(\nabla \mathbf{Y})$  in test. This inconsistency can be reconciled by  $\mathcal{L}_{\text{sd}}$ , as it uses  $\mathcal{W}(\nabla \mathbf{Y})$  for the input of  $\mathcal{F}$ .

## 4. Experiments

Performance evaluation is conducted on both simulated and real-world phase patterns. Following [22], we use NRMSE (Normalized Root Mean Squared Error) as the performance metric. As relative phase is the main focus in PU, the mean of unwrapped phase values is first aligned to that of the GT before the NRMSE calculation. Computational complexity is measured by three metrics: the number of parameters, the number of FLOPs (floating point of operations), and the average test time per sample of size  $256 \times 256$ .

**Implementation details:** Our proposed approach, called U3Net (U3 for Unsupervised, Unrolling and Unwrapping) for convenience, is implemented in Python and run on a single NVIDIA RTX 3080 GPU. We set  $\eta = 0.5$  in  $\mathcal{L}_{\text{total}}$  and initialize model weights via kaiming Uniform. The training employs the Adam optimizer with 500 epochs and a batch size of 10. The learning rate is initialized as  $1 \times 10^{-3}$  and exponentially decayed by a factor of 0.99 at every epoch.

**Methods for comparison:** We select eight representative methods of general PU as baselines, including LS [8], QGPU [11], Ryu. et al. [32], PhaseNet2.0 [35], PUDIP [47], SQD-LSTM [22], EESANet [49], and TriNet [37]. Additionally, two representative transformer NNs for general image restoration, UFormer [43] and Restormer [48], are also included for performance comparison. In all the baseline methods, LS and QGPU are conventional non-learning iterative methods, built upon least squares and path selection, respectively; PUDIP is an internal DL-based method; and the others are supervised DL-

| Dataset      |                    | MoGR        |             |             |             |             | RME         |             |             |             |             | #Param. | #FLOPs | Time     |
|--------------|--------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---------|--------|----------|
| SNR(dB)      |                    | 0           | 5           | 10          | 20          | 30          | 0           | 5           | 10          | 20          | 30          | (M)     | (G)    | (msec.)  |
| Non-Learning | LS                 | 7.91        | 2.28        | 1.20        | 0.37        | 0.15        | 7.85        | 2.50        | 1.25        | 0.40        | 0.12        | -       | -      | 8.39     |
| Supervised   | QGPU               | 17.12       | 2.12        | 1.15        | 0.36        | 0.11        | 17.26       | 2.29        | 1.20        | 0.39        | 0.12        | -       | -      | 14.94    |
|              | Ryu. <i>et al.</i> | 1.34        | 1.11        | 1.09        | 1.05        | 1.05        | 1.51        | 1.05        | 1.01        | 0.98        | 0.93        | 1.07    | 21.85  | 303.96   |
|              | PhaseNet2.0        | 8.35        | 8.19        | 8.07        | 8.06        | 8.04        | 9.29        | 8.53        | 7.53        | 6.95        | 6.86        | 1.15    | 11.93  | 20.87    |
|              | SQD-LSTM           | 0.87        | 0.71        | 0.70        | 0.69        | 0.68        | 1.57        | 1.13        | 1.12        | 1.10        | 1.09        | 0.90    | 4.07   | 13.04    |
|              | EESANet            | 0.78        | 0.77        | 0.76        | 0.76        | 0.74        | 1.31        | 1.31        | 1.25        | 1.24        | 1.06        | 61.68   | 75.27  | 9.85     |
|              | TriNet             | 5.65        | 5.55        | 5.46        | 5.40        | 5.34        | 5.40        | 5.33        | 5.21        | 5.10        | 5.05        | 13.61   | 65.48  | 11.16    |
|              | UFormer            | 0.50        | 0.47        | 0.47        | 0.46        | 0.45        | 0.58        | 0.51        | 0.49        | 0.49        | 0.48        | 20.60   | 40.98  | 42.00    |
|              | Restormer          | <b>0.45</b> | <u>0.38</u> | <u>0.36</u> | <u>0.36</u> | <u>0.35</u> | <b>0.50</b> | <u>0.43</u> | <u>0.42</u> | <u>0.41</u> | <u>0.41</u> | 3.02    | 17.23  | 44.69    |
| Unsupervised | PUDIP              | 17.53       | 15.16       | 7.87        | 0.34        | 0.11        | 13.10       | 7.22        | 2.62        | 0.38        | 0.12        | 2.33    | 21.58  | 99695.01 |
|              | U3Net              | 0.69        | <b>0.25</b> | <b>0.19</b> | <b>0.16</b> | <b>0.10</b> | 1.12        | <b>0.38</b> | <b>0.27</b> | <b>0.17</b> | <b>0.12</b> | 0.74    | 8.77   | 10.28    |

Table 1. Performance comparison on synthetic datasets in terms of NRMSE(%). **Bold** for **Best** and underline for second-best.

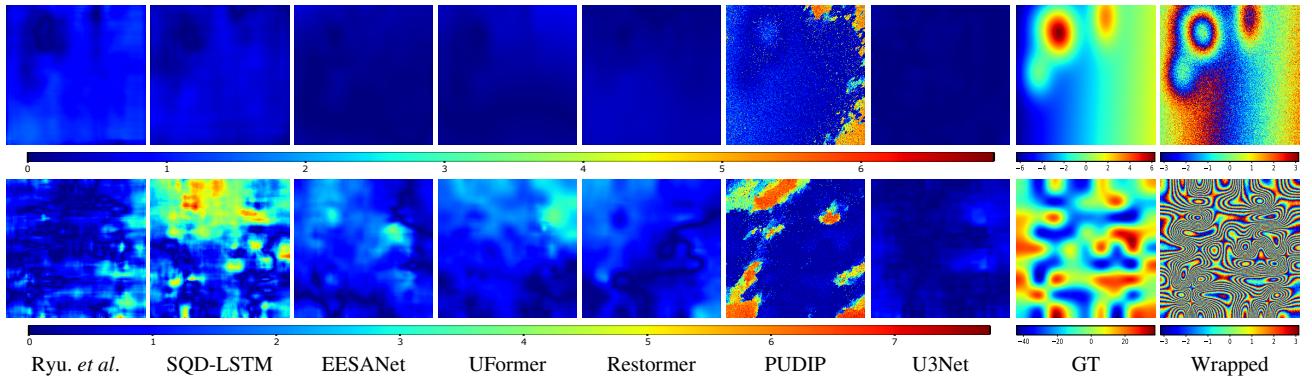


Figure 3. Visualization of PU results on MoGR (top) and RME (bottom), showing **residual images** to facilitate visual comparison.

based methods with their NNs retrained on our generated paired phase data. Specifically, as UFormer and Restormer are not originally designed for PU, we train their models using the supervised loss of SQD-LSTM [22] for better performance.

#### 4.1. Evaluation on Simulated Phase Patterns

**Data preparation:** Two types of simulated data are generated for training and evaluation. The first type follows [22], where a GT phase image of size  $256 \times 256$  is generated through the superposition of a mixture of Gaussians (MoG) with  $c$  clusters and randomly selected ramps. The phase range is defined within  $[-2p\pi, 2p\pi]$ . The resulting dataset is called MoGR (MoG with Ramp). The noisy wrapped phase images are synthesized via Eq. (1), with  $\mathcal{N}$  defined as Gaussian white noise. For each GT, only one noisy wrapped phase image is generated. With  $c$  and  $p$  uniformly selected from  $\{1, \dots, 4\}$  and  $\{1, \dots, 7\}$  respectively, we generate 5000 noisy wrapped phase images with SNR uniformly selected from  $\{0, 5, 10, 20, 30, 60\}$  to construct the training set. Using the same way, 1000 noisy wrapped phase images are generated for test with SNR in  $\{0, 5, 10, 20, 30\}$ .

The other dataset follows a different construction scheme

named random matrix enlargement [40], thus called RME dataset. Concretely, a GT phase image is formed through enlarging the square matrix of a distribution from an initial size  $s \times s$  to size  $256 \times 256$  via interpolation. The size parameter  $s$  is selected uniformly from  $\{2, \dots, 10\}$ , the distribution type is chosen from the  $\{\text{Uniform}, \text{Normal}\}$  distributions, and the interpolation method is chosen from the  $\{\text{Bilinear}, \text{Bicubic}\}$  interpolations. Then the phase image is re-scaled to the range varying in  $[-2p\pi, 2p\pi]$  with  $p$  randomly chosen from  $\{1, \dots, 7\}$ . The noisy wrapped phase images are obtained by the same way as the MoGR dataset.

**Results and analysis:** Table 1 lists the quantitative results on both datasets, including those in computational complexity.<sup>1</sup> Even trained without GT data, U3Net achieves the best results in 8/10 settings, using a lightweight model. Concretely, U3Net outperforms the two conventional non-learning-based methods, LS and QGPU. In comparison to PUDIP, an unsupervised internal DL-based method, U3Net also shows significant performance gain. The performance advantage of U3Net over these three GT-free methods be-

<sup>1</sup>PUDIP involves many forward passes for test-time learning. We report its #FLOPs in one forward pass and running time of the whole process.

| SNR(dB) | LS   | QGPU | Ryu. <i>et al.</i> | PhaseNet2.0 | SQD-LSTM | EESANet | TriNet | PU-GAN | PUNet | UFormer | Restormer   | PUDIP       | U3Net       |
|---------|------|------|--------------------|-------------|----------|---------|--------|--------|-------|---------|-------------|-------------|-------------|
| 5       | 3.31 | 3.13 | 1.41               | 2.28        | 1.76     | 2.45    | 5.04   | 13.73  | 9.59  | 1.46    | <u>1.06</u> | 9.99        | <b>1.00</b> |
| 10      | 1.84 | 1.96 | 1.27               | 1.69        | 1.52     | 1.99    | 4.69   | 11.84  | 9.20  | 1.28    | <u>0.93</u> | 5.30        | <b>0.82</b> |
| 20      | 0.94 | 1.13 | 1.24               | 1.43        | 1.48     | 1.79    | 4.46   | 11.62  | 9.01  | 0.97    | 0.91        | <u>0.47</u> | <b>0.46</b> |

Table 2. Performance comparison on InSAR phase data in terms of NRMSE(%). **Bold** for **Best** and underline for second-best.

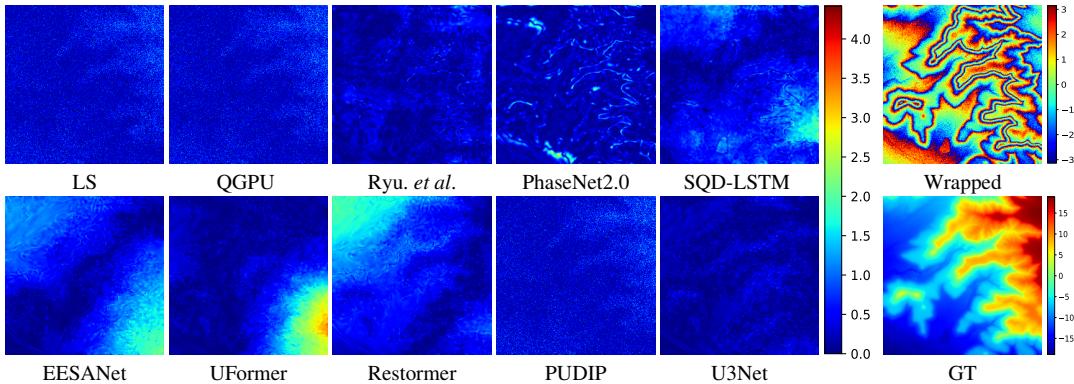


Figure 4. Visualization of unwrapped phase images on InSAR data, showing **residual images** to facilitate visual comparison.

comes more noticeable for lower SNR (*i.e.* heavier noise), particularly compared to PUDIP. This is because U3Net has an explicit mechanism to deal with noise during training, thereby achieving higher noise robustness. In addition, U3Net is much faster than PUDIP, due to that U3Net employs end-to-end pre-trained models, avoiding the time-costly per-sample training of PUDIP.

Surprisingly, U3Net also surpasses the supervised DL-based methods in many settings. In the lowest-SNR case, U3Net performs worse than UFormer and Restormer, two supervised transformer models. This is probably due to that severe noise increases the number of outlier points, lowering the effectiveness of the training function. Even that, U3Net still outperforms other supervised methods. The model of U3Net is the smallest among all the DL-based methods, indicating that the excellent performance of U3Net is mainly attributed to its interpretable physics-encoded NN architecture and its effective training scheme, rather than the use of larger models. Meanwhile, U3Net is computationally efficient, with the second smallest number of FLOPs and the third fastest inference speed.

See Fig. 3 for the visualization of PU results from different methods. The residuals to GTs produced by U3Net are much more insignificant compared to the internal DL-based PUDIP. In comparison to supervised methods, U3Net outperforms all of them on the image from RME dataset and shows comparable results on the image of MoGR dataset.

## 4.2. Evaluation on InSAR Phase Patterns

The InSAR phase data are generated by converting the digital elevation model at Texas, U.S. into absolute phases,

following [33]. Totally, 4000 (1000) GT phase images are obtained by cropping the  $256 \times 256$  patches from the absolute phases for training (test). Each phase value is scaled into  $[-6\pi, 6\pi]$ . The noisy wrapped phases are generated by adding Gaussian noise following the experiments in Sec. 4.1, and we test on three noise levels: 5, 10, 20. In addition to the previously-compared general PU methods and general-purpose transformer NNs, we also compare with two specific methods designed for InSAR PU: PUNet [44] and PU-GAN [57]. All the compared models are retrained using the same data as ours.

See Table 2 for the quantitative comparison on this InSAR dataset. Our U3Net ranks the first among all compared methods over all noise settings, noticeably surpassing other GT-free methods as well as many supervised methods in many cases. The visual comparison provided in Fig. 4 also shows the advantage of U3Net. All these results further verify the effectiveness of our approach in dealing with phase patterns from real scenes.

## 4.3. Ablation Studies

We form some baselines from U3Net for ablation studies, which are retrained and tested on the MoGR dataset. Their results are summarized in Table 3.

**Study on loss function:** We design the following schemes to analyze the loss function: (a)  $\mathcal{L}_{sr} \rightarrow \mathcal{L}$ : remove the outer  $\mathcal{W}$  applied to the residual, from the self-reconstruction loss  $\mathcal{L}_{sr}$ , *i.e.*, replacing  $\mathcal{L}_{sr}$  by the loss  $\mathcal{L}$  defined in Eq. (11); (b) w/o  $\nabla U$ : remove the noise re-corruption mechanism in  $\mathcal{L}_{sr}$  by discarding the  $\pm \nabla U$  terms; (c)  $\mathcal{L}_{sr} \rightarrow \mathcal{L}$  w/o  $\nabla U$ : combine the former cases, yielding a plain gradient loss

| Setting  | SNR=0 | SNR=5 | SNR=10 | SNR=20 | SNR=30 |
|--|-------|-------|--------|--------|--------|
| $\mathcal{L}_{\text{sr}} \rightarrow \mathcal{L}$                | 25.45 | 23.48 | 23.26  | 23.23  | 23.20  |
| w/o $\nabla U$   | 6.16  | 2.40  | 1.36   | 0.69   | 0.60   |
| $\mathcal{L}_{\text{sr}} \rightarrow \mathcal{L}$ w/o $\nabla U$ | 25.85 | 23.57 | 23.42  | 23.38  | 23.34  |
| w/o $\mathcal{L}_{\text{sd}}$                                    | 1.14  | 0.33  | 0.23   | 0.22   | 0.17   |
| Supervised   | 0.29  | 0.16  | 0.13   | 0.11   | 0.05   |
| w/o $E$  | 7.16  | 0.75  | 0.44   | 0.36   | 0.31   |
| AGD $\rightarrow$ GD   | 2.14  | 0.35  | 0.21   | 0.17   | 0.12   |
| w/o CAM  | 0.98  | 0.31  | 0.22   | 0.21   | 0.15   |
| $\mathcal{W}(\nabla Y) \rightarrow \nabla Y$                     | 2.88  | 1.74  | 1.50   | 1.46   | 1.39   |
| DUN $\rightarrow$ UFormer  | 1.55  | 1.41  | 1.31   | 1.30   | 1.30   |
| U3Net (original)   | 0.69  | 0.25  | 0.19   | 0.16   | 0.10   |

Table 3. Results in ablation studies in terms of NRMSE(%).

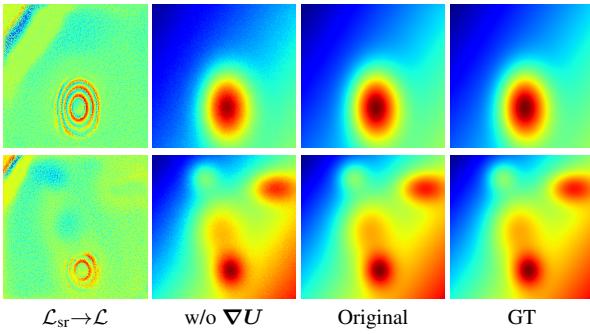


Figure 5. Visualization of unwrapped phase images using models trained by different loss functions.  $\mathcal{L}_{\text{sr}} \rightarrow \mathcal{L}$  and w/o  $\nabla U$  produce noticeable errors than Original. Zoom in for a better view.

$\|\nabla \mathcal{F}(\mathcal{W}(\nabla Y)) - \nabla Y\|_{\text{F}}^2$ ; (d) w/o  $\mathcal{L}_{\text{sd}}$ : remove the self-distillation loss  $\mathcal{L}_{\text{sd}}$ , only employing  $\mathcal{L}_{\text{sr}}$  for training; and (e) Supervised: train with a gradient-domain supervised loss  $\|\nabla \mathcal{F}(\mathcal{W}(\nabla Y)) - \nabla X\|_{\text{F}}^2$ .

In addition to Table 3, we also provide some visual comparisons in Fig. 5 for demonstration. From both the quantitative and qualitative results, we have following observations. (a) The outer  $\mathcal{W}$  in  $\mathcal{L}_{\text{sr}}$  is critical to the performance of U3Net. (b) The re-corruption mechanism in  $\mathcal{L}_{\text{sr}}$  does help eliminate the measurement noise. (c) A simple gradient loss function does not work, with further performance decrease. (d) While the self-reconstruction loss already yields promising results, the self-distillation loss benefits further performance gain. (e) U3Net performs closely to its supervised counterpart when  $\text{SNR} \geq 5$ . We also see that U3Net has its supervised counterpart outperforming the supervised UFormer reported in Table 1, demonstrating its superiority.

**Study on DUN:** To analyze the components in the DUN architecture of U3Net, we design the following baseline models: (a) w/o  $E$ : abandon the error term  $E$  during unrolling, forming a DUN alternating the accelerated gradient descent step and the sub-NN for estimating  $X^{(t)}$ ; (b) AGD $\rightarrow$ GD: replace the Nesterov’s AGD steps by one-step standard gradient; (c) w/o CAM: remove the CAM from U3Net; (d)

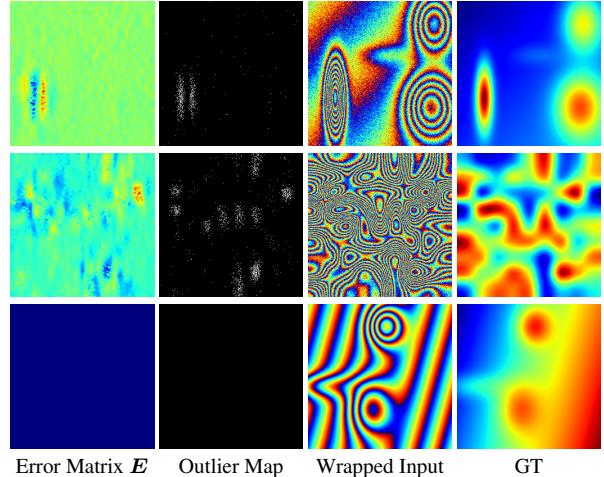


Figure 6. Visualization of error matrices  $E$  estimated from three wrapped phase images. The outlier maps use white pixels to indicate the points  $[x, y]$  satisfying  $\|\nabla X[x, y] + \nabla N[x, y]\|_{\infty} \geq \pi$  and black pixels otherwise.

$\mathcal{W}(\nabla Y) \rightarrow \nabla Y$ : use  $\nabla Y$  to replace  $\mathcal{W}(\nabla Y)$  in the input of U3Net as well in the total loss; (e) DUN $\rightarrow$ UFormer: replace the DUN by UFormer, keeping the loss unchanged.

As demonstrated in Table 3, each component in the DUN of U3Net is useful. (a) One critical part is the introduction of the error matrix  $E$  for unrolling. It brings significant performance gain, particularly in dealing with low-SNR data, as the absorption of outliers breaking the Itoh’s condition is more desired for heavier noise. See Fig. 6 for a visualization of estimated error matrices, which successfully identify the outliers. (b)&(c) The AGD and CAM have similar contribution to the performance of U3Net. (d) Using  $\mathcal{W}(\nabla Y)$  for DUN is also crucial, serving as a better choice compared to using  $\nabla Y$ . (e) Our proposed loss function also works for UFormer, but yielding worse results than U3Net, which indicates the higher effectiveness of our physics-encoded DUN for unsupervised PU, compared to general transformer models.

## 5. Conclusion

This work proposed a fully unsupervised DL approach for PU, eliminating the dependence on GTs in end-to-end training. Utilizing the connection between (wrapped) gradients of wrapped phases and gradients of original phases, we constructed a physics-encoded DUN, trained via a noise-resistant self-reconstruction loss complemented by a self-distillation loss. Empirical evaluations confirmed that our approach surpassed existing GT-free methods and rivaled supervised counterparts. Similar to current unsupervised and synthetic data-dependent supervised techniques, our approach presumes known noise statistics. Future research will focus on dealing with unknown noise profiles.

## References

- [1] Liheng Bian, Xin Wang, Daoyu Li, Qiuling Ren, and Dezhi Zheng. Robust phase unwrapping via non-local regularization. *Optics Letters*, 48(6):1399–1402, 2023. 1, 2
- [2] David Blinder, Heidi Ottavaere, Adrian Munteanu, and Peter Schelkens. Efficient multiscale phase unwrapping methodology with modulo wavelet transform. *Optics Express*, 24(20):23094–23108, 2016. 1, 2
- [3] Donald J Bone. Fourier fringe analysis: the two-dimensional phase unwrapping problem. *Applied Optics*, 30(25):3627–3632, 1991. 1, 2
- [4] Mingqin Chen, Peikang Lin, Yuhui Quan, Tongyao Pang, and Hui Ji. Unsupervised phase retrieval using deep approximate mmse estimation. *IEEE Transactions on Signal Processing*, 70:2239–2252, 2022. 3
- [5] Mingqin Chen, Yuhui Quan, Tongyao Pang, and Hui Ji. Non-blind image deconvolution via leveraging model uncertainty in an untrained deep neural network. *International Journal of Computer Vision*, 130(7):1770–1789, 2022. 3
- [6] Gili Dardikman and Natan T Shaked. Phase unwrapping using residual neural networks. In *Computational Optical Sensing and Imaging*, pages CW3B–5. Optica Publishing Group, 2018. 3
- [7] Qiaoqiao Ding, Gaoyu Chen, Xiaoqun Zhang, Qiu Huang, Hui Ji, and Hao Gao. Low-dose ct with deep learning regularization via proximal forward–backward splitting. *Physics in Medicine & Biology*, 65(12):125009, 2020. 3
- [8] Dennis C Ghiglia and Louis A Romero. Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. *Journal of the Optical Society of America A*, 11(1):107–117, 1994. 2, 5
- [9] Richard M Goldstein, Howard A Zebker, and Charles L Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23(4):713–720, 1988. 1, 2
- [10] Michał Gontarz, Vibekananda Dutta, Małgorzata Kujańska, and Wojciech Krauze. Phase unwrapping using deep learning in holographic tomography. *Optics Express*, 31(12):18964–18992, 2023. 1, 3
- [11] Miguel Arevalillo Herráez, David R Burton, Michael J Lalor, and Munther A Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Applied Optics*, 41(35):7437–7444, 2002. 1, 2, 5
- [12] Howard YH Huang, L Tian, Z Zhang, Y Liu, Z Chen, and G Barbastathis. Path-independent phase unwrapping using phase gradient and total-variation (tv) denoising. *Optics Express*, 20(13):14075–14089, 2012. 1, 2
- [13] Kazuyoshi Itoh. Analysis of the phase unwrapping algorithm. *Applied Optics*, 21(14):2470–2470, 1982. 2, 3
- [14] Charles VJ Jakowatz, Daniel E Wahl, Paul H Eichel, Dennis C Ghiglia, and Paul A Thompson. *Spotlight-mode synthetic aperture radar: a signal processing approach*. Springer Science & Business Media, 2012. 1
- [15] Ulugbek S Kamilov, Ioannis N Papadopoulos, Morteza H Shoreh, Demetri Psaltis, and Michael Unser. Isotropic inverse-problem approach for two-dimensional phase unwrapping. *Journal of the Optical Society of America A*, 32(6):1092–1100, 2015. 1, 2
- [16] Thomas Kreis. *Handbook of holographic interferometry: optical and digital methods*. John Wiley & Sons, 2006. 1
- [17] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019. 3
- [18] Ji Li, Weixi Wang, Yuesong Nan, and Hui Ji. Self-supervised blind motion deblurring with deep expectation maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13986–13996, 2023. 3
- [19] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noiser2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12064–12072, 2020. 2
- [20] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17399–17410, 2022. 3
- [21] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2043–2052, 2021. 2, 3
- [22] Malsha V Perera and Ashwin De Silva. A joint convolutional and spatial quad-directional lstm network for phase unwrapping. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4055–4059. IEEE, 2021. 1, 3, 5, 6
- [23] Mark D Pritt. Phase unwrapping by means of multigrid techniques for interferometric sar. *IEEE Transactions on Geoscience and Remote Sensing*, 34(3):728–738, 1996. 1, 2
- [24] Xinran Qin, Yuhui Quan, and Hui Ji. Enhanced deep unrolling networks for snapshot compressive hyperspectral imaging. *Neural Networks*, 174:106250, 2024. 3
- [25] Yi Qin, Shujia Wan, Yuhong Wan, Jiawen Weng, Wei Liu, and Qiong Gong. Direct and accurate phase unwrapping with deep neural network. *Applied Optics*, 59(24):7258–7267, 2020. 3
- [26] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1890–1898, 2020. 3
- [27] Yuhui Quan, Zhuojie Chen, Huan Zheng, and Hui Ji. Learning deep non-blind image deconvolution without ground truths. In *European Conference on Computer Vision*, pages 642–659. Springer, 2022. 3
- [28] Yuhui Quan, Xinran Qin, Tongyao Pang, and Hui Ji. Dual-domain self-supervised learning and model adaption for deep compressive imaging. In *European Conference on Computer Vision*, pages 409–426. Springer, 2022. 3
- [29] Yuhui Quan, Zhile Chen, Tongyao Pang, and Hui Ji. Unsupervised deep learning for phase retrieval via teacher-student

- distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 3
- [30] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 5
- [31] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020. 3
- [32] Kanghyun Ryu, Sung-Min Gho, Yoonho Nam, Kevin Koch, and Dong-Hyun Kim. Development of a deep learning method for phase unwrapping mr images. In *Proceedings of the International Society for Magnetic Resonance in Medicine*, page 4707, 2019. 3, 5
- [33] Francescopaolo Sica, Francesco Calvanese, Giuseppe Scarpa, and Paola Rizzoli. A cnn-based coherence-driven approach for insar phase unwrapping. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2020. 7
- [34] GE Spoorthi, Subrahmanyam Gorthi, and Rama Krishna Sai Subrahmanyam Gorthi. Phasenet: A deep convolutional neural network for two-dimensional phase unwrapping. *IEEE Signal Processing Letters*, 26(1):54–58, 2018. 3
- [35] GE Spoorthi, Rama Krishna Sai Subrahmanyam Gorthi, and Subrahmanyam Gorthi. Phasenet 2.0: Phase unwrapping of noisy data based on deep learning approach. *IEEE Transactions on Image Processing*, 29:4862–4872, 2020. 3, 5
- [36] Xianyu Su and Wenjing Chen. Reliability-guided phase unwrapping algorithm: a review. *Optics and Lasers in Engineering*, 42(3):245–261, 2004. 1, 2
- [37] Krishna Sumanth, Vaishnavi Ravi, and Rama Krishna Gorthi. A multi-task learning for 2d phase unwrapping in fringe projection. *IEEE Signal Processing Letters*, 29:797–801, 2022. 5
- [38] Julián Tachella, Dongdong Chen, and Mike Davies. Unsupervised learning from incomplete measurements for inverse problems. *Advances in Neural Information Processing Systems*, 35:4983–4995, 2022. 3
- [39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 3, 5
- [40] Kaiqiang Wang, Ying Li, Qian Kemao, Jianglei Di, and Jianlin Zhao. One-step robust deep learning phase unwrapping. *Optics Express*, 27(10):15100–15115, 2019. 3, 6
- [41] Kaiqiang Wang, Qian Kemao, Jianglei Di, and Jianlin Zhao. Deep learning spatial phase unwrapping: a comparative review. *Advanced Photonics Nexus*, 1(1):014001, 2022. 1, 2
- [42] Weixi Wang, Ji Li, and Hui Ji. Self-supervised deep image restoration via adaptive stochastic gradient langevin dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2022. 3
- [43] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 5
- [44] Zhipeng Wu, Teng Wang, Yingjie Wang, Robert Wang, and Daqing Ge. Deep learning for the detection and phase unwrapping of mining-induced deformation in large-scale interferograms. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2021. 7
- [45] Xianming Xie. Iterated unscented kalman filter for phase unwrapping of interferometric fringes. *Optics Express*, 24(17):18872–18897, 2016. 1, 2
- [46] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing*, 29:9316–9329, 2020. 2
- [47] Fangshu Yang, Thanh-An Pham, Nathalie Brandenberg, Matthias P Lütolf, Jianwei Ma, and Michael Unser. Robust phase unwrapping via deep image prior for quantitative phase imaging. *IEEE Transactions on Image Processing*, 30:7025–7037, 2021. 1, 2, 3, 5
- [48] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 5
- [49] Junkang Zhang and Qingguang Li. Eesanet: edge-enhanced self-attention network for two-dimensional phase unwrapping. *Optics Express*, 30(7):10470–10490, 2022. 3, 5
- [50] Junchao Zhang, Xiaobo Tian, Jianbo Shao, Haibo Luo, and Rongguang Liang. Phase unwrapping in optical metrology via denoised and convolutional segmentation networks. *Optics Express*, 27(10):14903–14912, 2019. 3
- [51] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020. 3, 5
- [52] Teng Zhang, Shaowei Jiang, Zixin Zhao, Krishna Dixit, Xiaofei Zhou, Jia Hou, Yongbing Zhang, and Chenggang Yan. Rapid and robust two-dimensional phase unwrapping via deep learning. *Optics Express*, 27(16):23173–23185, 2019. 3
- [53] Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. Amp-net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2020. 3
- [54] Ming Zhao, Lei Huang, Qican Zhang, Xianyu Su, Anand Asundi, and Qian Kemao. Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies. *Applied Optics*, 50(33):6214–6224, 2011. 2
- [55] Hongyi Zheng, Hongwei Yong, and Lei Zhang. Deep convolutional dictionary learning for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 630–641, 2021. 3
- [56] Hongyu Zhou, Chuanli Cheng, Hao Peng, Dong Liang, Xin Liu, Hairong Zheng, and Chao Zou. The phu-net: A robust phase unwrapping method for mri based on deep learning.

*Magnetic Resonance in Medicine*, 86(6):3321–3333, 2021.

[1](#), [3](#)

- [57] Lifan Zhou, Hanwen Yu, Vito Pascazio, and Mengdao Xing. Pu-gan: a one-step 2-d insar phase unwrapping based on conditional generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–10, 2022. [1](#), [3](#), [7](#)