Review Article

ADVANCED
PHOTONICS
NEXUS

Deep learning spatial phase unwrapping: a comparative review

Kaiqiang Wang,[a,b] **Qian Kemao**,[c,*] **Jianglei Di**,[a,b,d,*] **and Jianlin Zhao**[a,b,*]

[a]Northwestern Polytechnical University, School of Physical Science and Technology, Shaanxi Key Laboratory of Optical Information Technology, Xi'an, China
[b]Ministry of Industry and Information Technology, Key Laboratory of Light Field Manipulation and Information Acquisition, Xi'an, China
[c]Nanyang Technological University, School of Computer Science and Engineering, Singapore
[d]Guangdong University of Technology, Guangdong Provincial Key Laboratory of Photonics Information Technology, Guangzhou, China

Abstract. Phase unwrapping is an indispensable step for many optical imaging and metrology techniques. The rapid development of deep learning has brought ideas to phase unwrapping. In the past four years, various phase dataset generation methods and deep-learning-involved spatial phase unwrapping methods have emerged quickly. However, these methods were proposed and analyzed individually, using different strategies, neural networks, and datasets, and applied to different scenarios. It is thus necessary to do a detailed comparison of these deep-learning-involved methods and the traditional methods in the same context. We first divide the phase dataset generation methods into random matrix enlargement, Gauss matrix superposition, and Zernike polynomials superposition, and then divide the deep-learning-involved phase unwrapping methods into deep-learning-performed regression, deep-learning-performed wrap count, and deep-learning-assisted denoising. For the phase dataset generation methods, the richness of the datasets and the generalization capabilities of the trained networks are compared in detail. In addition, the deep-learning-involved methods are analyzed and compared with the traditional methods in ideal, noisy, discontinuous, and aliasing cases. Finally, we give suggestions on the best methods for different situations and propose the potential development direction for the dataset generation method, neural network structure, generalization ability enhancement, and neural network training strategy for the deep-learning-involved spatial phase unwrapping methods.

Keywords: phase unwrapping; deep learning; interferometry.

Received Jun. 22, 2022; accepted for publication Jun. 28, 2022; published online Aug. 3, 2022.

[DOI: 10.1117/1.APN.1.1.014001]

1 Introduction

Estimation of absolute (true) phase is an important but challenging problem in many imaging or measurement techniques, such as optical interferometry (OI),[1] magnetic resonance imaging (MRI),[2] fringe projection profilometry (FPP),[3,4] and interferometric synthetic aperture radar (InSAR)[5,6] (see Fig. 1).

The importance lies in that the estimated absolute phase of the aforementioned applications is directly proportional to the desired physical quantities such as the distribution of thickness or refractive index for OI, the distribution of magnetic susceptibility or the velocity of blood flow for MRI, the three-dimensional (3D) distribution of the object surface for FPP, and the surface height of the topography or ground deformation for InSAR.

The challenge lies in that the initial phase of the aforementioned applications is limited in the range of $(-\pi, \pi]$ since it is calculated from the complex amplitude field (CAF) by the arctangent function. However, in most cases, the phase range corresponding to the sample exceeds this limitation. Therefore, to obtain the desired physical quantities, the absolute phase must be estimated from the initial wrapped phase that is the so-called phase unwrapping.

The wrapped phase φ and absolute phase ψ have the following relationship:

$$\psi(\boldsymbol{r}) = \varphi(\boldsymbol{r}) + 2\pi k(\boldsymbol{r}), \tag{1}$$

*Address all correspondence to Qian Kemao, mkmqian@ntu.edu.sg; Jianglei Di,, jiangleidi@nwpu.edu.cn; Jianlin Zhao, jlzhao@nwpu.edu.cn

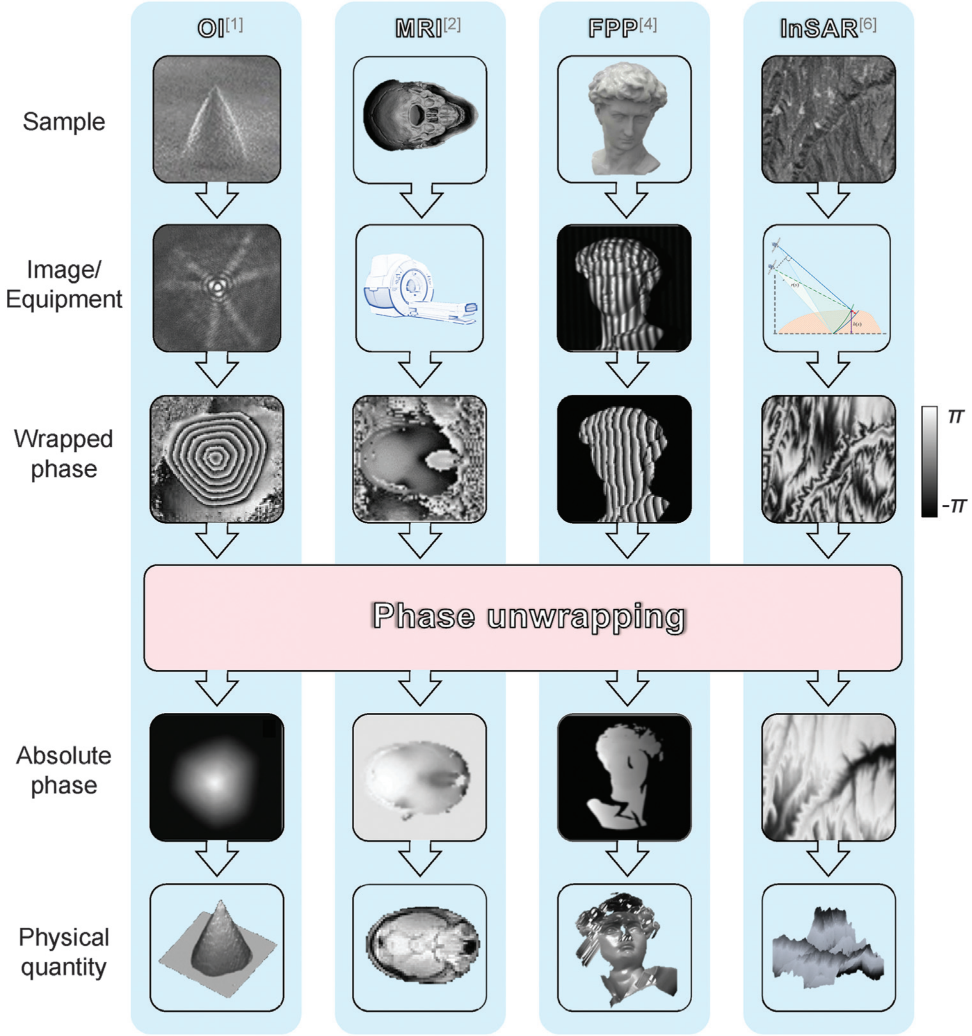


Fig. 1 Phase unwrapping in OI,[1] MRI,[2] FPP,[4] and InSAR.[6]

where $\boldsymbol{r}$ is the vector coordinate and k is the wrap count, which is an integer.

Spatial phase unwrapping, i.e., getting ψ solely by φ, is straightforward but ill-posed, because both ψ and k are unknown in Eq. (1). However, if the absolute phase is continuous, i.e., the Itoh condition is satisfied, the problem becomes well-posed.[7] In the Itoh condition, ψ and φ satisfy the following relationship:

$$\nabla\psi = W[\nabla\varphi] = \begin{cases} \nabla\varphi, & |\nabla\varphi| \leq \pi \\ \nabla\varphi - 2\pi, & \nabla\varphi > \pi \\ \nabla\varphi + 2\pi, & \nabla\varphi < -\pi \end{cases}, \tag{2}$$

where $W[\cdot]$ is the wrap operator that removes multiples of 2π so that the output is within $[-\pi, \pi)$ and ∇ is the difference operator. Then, the absolute phase can be calculated as

$$\psi(\boldsymbol{r}_1) = \psi(\boldsymbol{r}_0) + \int_L \nabla\psi \mathrm{d}\boldsymbol{r} = \psi(\boldsymbol{r}_0) + \int_L W[\nabla\varphi]\mathrm{d}\boldsymbol{r}, \tag{3}$$

where $\boldsymbol{r}_0$ is the starting point, $\boldsymbol{r}_1$ is the current point, and L is an arbitrary integration path linking the two points. The simplest integration path is the line-by-line scan.

Unfortunately, in practical applications, noise, discontinuity and aliasing violate the Itoh condition and fail the phase unwrapping. Thus, in the traditional spatial phase unwrapping methods, the path-following method determines better integration paths by branch cuts,[8] quality maps,[9] etc., to avoid the influence of these invalid pixels, and the optimization-based method obtains the absolute phase by minimizing the difference between absolute phase gradients and wrapped phase gradients, as follows:

$$\psi_O = \arg\min_{\psi_t}\left[\iint f(\nabla\psi_t - W[\nabla\varphi])\mathrm{d}A\right], \tag{4}$$

where ψ_t is the absolute phase field to be optimized, and its optimal value is denoted as ψ_O; $f(\cdot)$ is an objective function such as energy functions[10] and L_p-norm.[11] In addition to putting all efforts into improving the unwrapping algorithm, window Fourier transform (WFT) or other filters can be used to filter the noise pixels, with the result denoted as ψ_F, before phase unwrapping.[12–14] The filtered phase is then unwrapped into ψ_F. Note that ψ_O from optimization and ψ_F with prefiltering are not the same as ψ, thus do not satisfy Eq. (1). Thus, the congruence operation is sometimes applied as[14,15]

$$\psi_C = \psi_M + W[\varphi - \psi_M], \tag{5}$$

where ψ_M represents any successful but not congruent phase unwrapping results of a particular method, such as ψ_F and ψ_O. More details and comparison of the traditional spatial phase unwrapping methods can be found in the classic books and review papers.[5,16–19]

In fact, the purpose of the traditional spatial phase unwrapping methods is to avoid the negative impact of invalid points as much as possible, and it is mainly suitable for nonserious noise and some discrete discontinuous or aliasing points. In some extreme cases such as the presence of severe noise or locally isolated discontinuous or aliasing regions, the traditional methods will become ineffective.

Jin et al.[20] first proposed the use of deep learning to solve illposed inverse problems in imaging. This idea of using a deep neural network to learn the mapping relationship from input space to output space from paired datasets makes it possible to solve phase unwrapping in the aforementioned extreme cases. Thus, in the past 4 years, many deep-learning-involved methods have quickly emerged for spatial phase unwrapping that are still effective in the cases of noise, discontinuity, and aliasing due to not being constrained by the Itoh condition.[21–45]

These deep-learning-involved methods use different strategies to achieve phase unwrapping by supervised optimization of neural networks with specific datasets:

- Deep-learning-performed regression method (dRG) estimates the absolute phase directly from the wrapped phase by the neural network.[22–33] The used dataset contains the paired wrapped phase as input and absolute phase as ground truth (GT), as shown in Fig. 2(a).
- Deep-learning-performed wrap count method (dWC) first estimates the wrap count from the wrapped phase by the neural network, and then obtains the absolute phase by Eq. (1).[34–44] The used dataset contains the paired wrapped phase as input and wrap count as GT, as shown in Fig. 2(b).
- Deep-learning-assisted denoising method (dDN) first denoises the real and imaginary parts of CAF by the neural network, and then unwraps the pure (noise-free) wrapped phase using a traditional method.[45] The used dataset contains the paired noisy and pure real and imaginary parts of CAF as input and GT, respectively, as shown in Fig. 2(c).

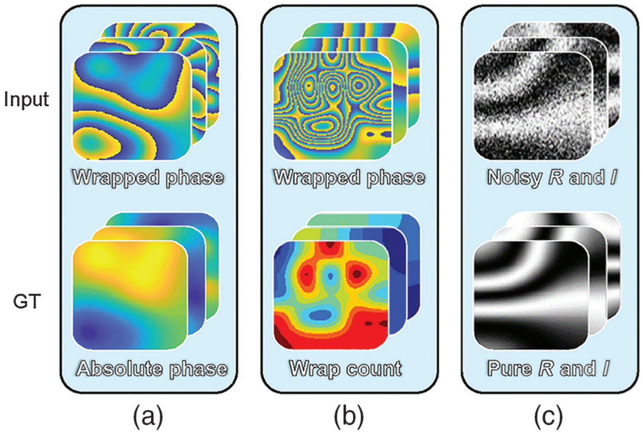


Fig. 2 Datasets of the deep-learning-involved phase unwrapping methods, for (a) dRG, (b) dWC, and (c) dDN. "*R*" and "*I*" represent the real and imaginary parts of CAF, respectively.

In addition, the absolute phase dataset generation methods used in deep-learning-involved methods can be divided into the following four categories:

- Random matrix enlargement (RME): The absolute phase of various distributions is obtained by enlarging a small random matrix of random size and range.[24,28,32]
- Gaussian functions superposition (GFS): The absolute phase of various distributions is obtained by weighted superposition of multiple Gaussian functions with different mean values and variances.[29,35,38,39,42–44]
- Zernike polynomials superposition (ZPS): The absolute phase of various distributions is obtained by weighted superposition of Zernike polynomials in different orders.[36,37,40,41,45]
- Real data reprocessing (RDR): The absolute phase of real samples is obtained by traditional methods.[23,27,30,31,33]

As shown in Fig. 3, the overall process of these deep-learning-involved phase unwrapping methods can be summarized as follows:

(a) The datasets including input and GT are generated through computer simulations or real experiments.
(b) Then through the training process, a one-time effort, the weights and biases of the neural network are adjusted by minimizing the loss function between the neural network output and GT.[46]
(c) After the training process, the neural network can perform phase unwrapping directly or indirectly.

Since these methods are proposed individually using different phase unwrapping strategies and dataset generation methods and applied to different scenarios, a comprehensive cross-comparison and a comparison with traditional methods are lacking, which obscures the true potential of deep-learning-involved phase unwrapping and dataset generation methods. In the present review, Sec. 2 offers a summary and classification of deep-learning-involved phase unwrapping methods; in Sec. 3, the dataset generation methods are summarized and classified, their performance and characteristics are compared, and the rules and tips of dataset generation are given; in Sec. 4, using the same dataset, the performance of the deep-learning-involved and traditional phase unwrapping methods in ideal, noisy, discontinuous, aliasing, and mixed cases is compared and summarized; in Sec. 5, the advantages and limitations of the existing deep-learning-involved phase unwrapping methods are summarized, and a deep learning phase unwrapping idea with joint supervision of dataset and physical-model is further proposed and demonstrated.

To give interested readers a quick start, we present a step-by-step guide to applying deep learning to phase unwrapping in the Supplemental Material, with dRG and RME as an example.

2 Phase Unwrapping Methods with Deep Learning

2.1 *Deep-Learning-Performed Regression (dRG) Method*

Phase unwrapping can be treated as a regression problem in which a neural network directly learns the mapping relationship between the wrapped phase and the absolute phase.[22–33] As illustrated in Fig. 4, after being fed with a wrapped phase, the trained network directly outputs the unwrapped (absolute) phase. Such a mapping relationship is most straightforward and intuitive, but

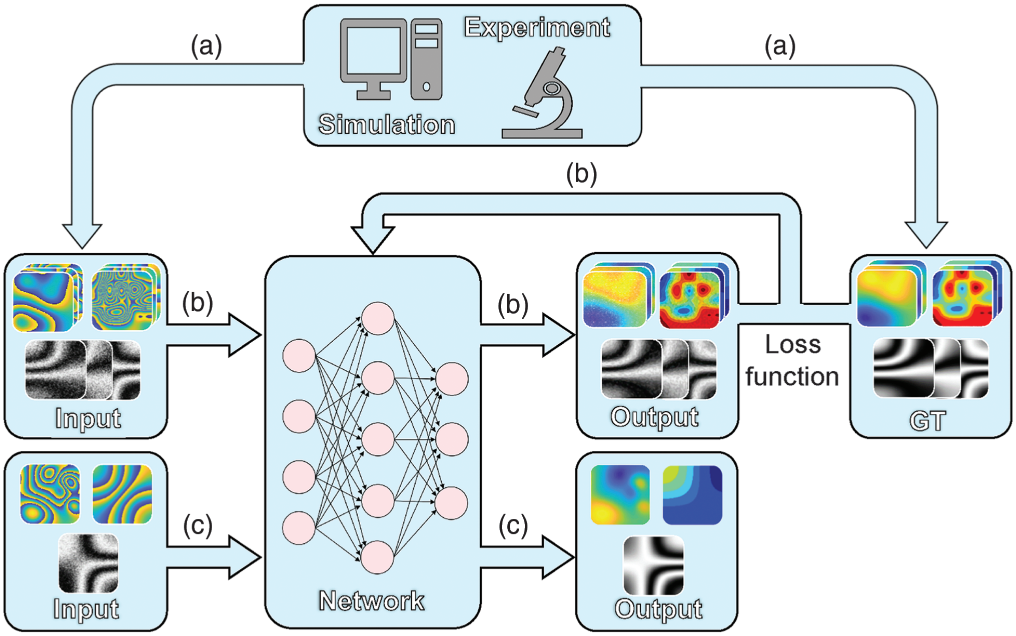


Fig. 3 Overall process of deep-learning-involved phase unwrapping methods.

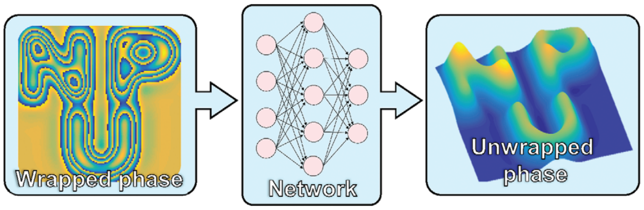


Fig. 4 Illustration of the dRG method.

the unwrapped phase does not strictly follow the relationship in Eq. (1). In other words, the unwrapped phase is incongruent, i.e., each pixel has a small error. The congruence operation in Eq. (5) can be applied if necessary.

Dardikman et al.[22,47] introduced the dRG method for simulated steep cells directly through a residual-block-based convolutional neural network (CNN), verified the dRG method with the congruence for real cells and compared the dRG method with the deep-learning-performed wrap count method.[23] In 2019, we proposed a phase dataset generation method in which we tested the trained network on real samples, analyzed the network generalization ability by the middle layer visualization, and verified the superiority of the dRG method in noisy and aliasing cases by comparing it with the traditional methods.[24,48] He et al.[25] and Ryu et al.[26] tested the phase unwrapping performance of the bidirectional recurrent neural network (RNN) and 3D-ResNet with MRI data. Dardikman-Yoffe et al.[27] open-sourced their real sample dataset and verified that the congruence could improve the accuracy and robustness of the dRG method in the case of a small number of wrap counts. Qin et al.[28] used a larger-capacity Res-UNet to obtain higher accuracy and proposed two evaluation indices. Perera and De Silva[29] and Park et al.[30] tested the phase unwrapping performance of the long short-term memory (LSTM) network and generative adversarial network (GAN). Zhou et al.[31] improved the robustness and efficiency of deep learning phase unwrapping by adding preprocessing and postprocessing. Xu et al.[32] improved the accuracy and robustness of phase unwrapping in an end-to-end case by using a composite loss function and adding more skip connections to Res-UNet. Zhou et al.[33] used the GAN in InSAR phase unwrapping.

2.2 Deep-Learning-Performed Wrap Count (dWC) Method

According to Eq. (1), if the wrap counts $k(x,y)$ are successfully determined, the phase unwrapping is also fulfilled by adding $2\pi k(x,y)$ to the wrapped phase, as illustrated in Fig. 5. Although the phase values are different from pixel to pixel, the wrap count is the same in one fringe period. Thus, phase unwrapping can be interestingly treated as a classification/segmentation problem in which a neural network predicts the wrap count for each pixel from the wrapped phase.[34–44] Such a mapping relationship is polarized, i.e., it is either correct or incorrect for each pixel.

This idea was introduced by Liang et al.[34] and Spoorthi et al.[35] Spoorthi et al.[35] proposed a phase dataset generation method and used the generated dataset to train a SegNet to predict the wrap count, which was postprocessed by clustering-based smoothness to alleviate the classification imbalance. Zhang et al.[36] performed phase unwrapping by three networks sequentially for wrapped phase denoising, wrap count predicting, and postprocessing, respectively. Zhang et al.[37,49] verified the performance of the network DeepLab-V3+ in the dWC method and proposed using refinement for postprocessing. Wu et al.[38,50] enhanced the simulated phase dataset by adding the noise from real data and used the multiscale context and the full resolution residual block (FRRes-UNet) to further optimize the UNet in the Doppler optical coherence tomography. Also, Spoorthi et al. improved the prediction accuracy of the wrap count of the method in Ref. 35 by introducing the priori-knowledge of absolute phase values and gradients into the loss function.[39] Zhao et al.[40] used a residual autoencoder network (RAEN) to predict the wrap

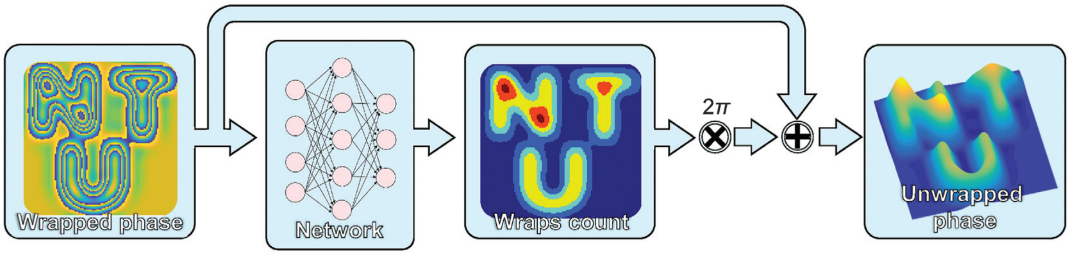


Fig. 5 Illustration of the dWC method.

counts, used an image-analysis-based postprocessed method to alleviate the classification imbalance, and adopted iterative-closest-point phase data stitching method to realize dynamic resolution. Zhu et al.[41] used the dWC method with postprocessing to do phase unwrapping in inertial confinement fusion target interferometric measurement. Vengala et al.[42,43,51] used the Y-Net to reconstruct the wrap count and pure wrapped phase at the same time. Zhang and Li[44] added atrous spatial pyramid pooling, positional self-attention, and edge-enhanced block to the neural network to get a higher accuracy and stronger robustness.

2.3 Deep-Learning-Assisted Denoising (dDN) Method

As mentioned in Sec. 1, in the traditional methods, we can either use complicated algorithms to directly unwrap or circumvent the invalid pixels or filter the invalid pixels before using the simplest algorithm to do phase unwrapping.[7,13,14] For the latter approach, the neural network can also be used for the invalid pixels filtering.[45]

Yan et al.[45] proposed using a neural network to denoise the real and imaginary parts of CAF for FPP. Then they used the line-scanning method to do phase unwrapping on the noise-filtered wrapped phase. In their work, denoising was performed by extracting the noise components from the noisy real and imaginary parts. After verification, we found that using neural networks to directly extract the pure real and imaginary components had the same performance. Thus, here, we adopt the direct way as shown in Fig. 6. In addition to this way of denoising directly using the neural network, the input of the neural network (the noisy real and imaginary parts) can also be semantically segmented to distinguish the sample area and the background area. Then the semantic segmentation results can be used as prior knowledge to make the neural network focus on the sample area.

2.4 Summary of the Deep-Learning-Involved Methods

For clarity, we summarize all the deep-learning-involved phase unwrapping methods mentioned above in Table 1, where "RME," "GFS," "ZPS," and "RDR" indicate the dataset generation methods, which will be introduced in Sec. 3. In this table, except for the phase unwrapping methods introduced above (column 1), we also include the network structures (column 5), the training datasets (column 6), and the loss functions (column 7), which directly affect the final effect of phase unwrapping.

The network structure is not within the scope of the comparison of this paper, so we choose Res-UNet as a unified network structure that has been widely used in different optical applications with excellent performance.[51–77] Furthermore, the loss functions can be incorporated into the networks as a variable. The dRG and dDN methods usually use mean squared error (MSE) or mean absolute error (MAE) as whole or main components of the loss function, while the dWC method usually uses cross-entropy (CE) as the whole or main component of the loss function. Thus, we use MAE as the loss function of the dRG and dDN, and use CE and MAE as the composite loss function, since which can effectively improve the network accuracy.[39]

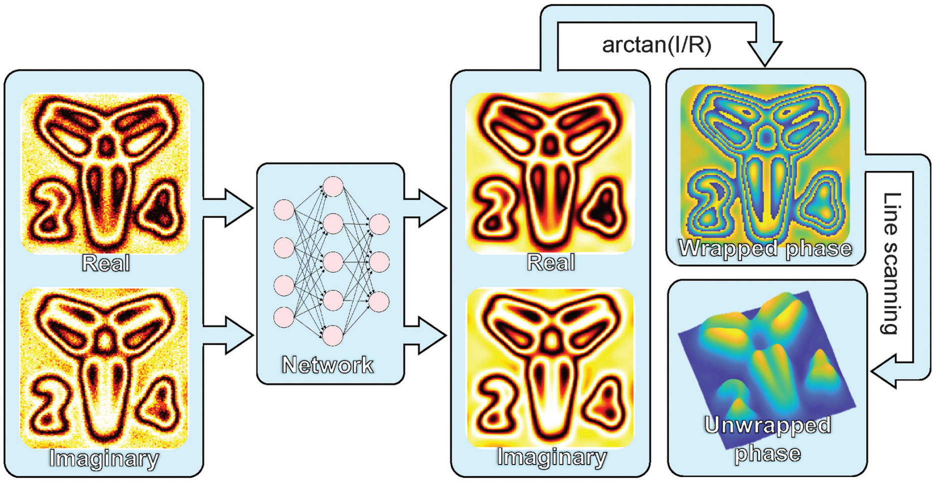


Fig. 6 Illustration of the dDN method.

Table 1 Summary of deep-learning-involved phase unwrapping methods. "—" indicates "not available."

Method	Date	Author	Ref.	Network	Dataset	Loss function
dRG	2018	Dardikman and Shaked	22	—	—	—
		Dardikman et al.	23	ResNet	RDR	MSE
	2019	Wang et al.	24	Res-UNet	RME	MSE
		He et al.	25	3D-ResNet	—	—
		Ryu et al.	26	RNN	—	Total variation + error variation
	2020	Dardikman-Yoffe et al.	27	Res-UNet	RDR	MSE
		Qin et al.	28	Res-UNet	RME	MAE
	2021	Perera and De Silva	29	LSTM	GFS	Total variation + error variation
		Park et al.	30	GAN	RDR	MAE + adversarial loss
		Zhou et al.	31	UNet	RDR	MAE + residues
	2022	Xu et al.	32	MNet	RME	MAE and MS-SSIM
		Zhou et al.	33	GAN	RDR	MAE + adversarial loss
dWC	2018	Liang et al.	34	—	—	—
		Spoorthi et al.	35	SegNet	GFS	CE
	2019	Zhang et al.	36	UNet	ZPS	CE
		Zhang et al.	37	DeepLab-V3+	ZPS	CE
	2020	Wu et al.	38	FRRes-UNet	GFS	CE
		Spoorthi et al.	39	Dense-UNet	GFS	MAE + residues + CE
		Zhao et al.	40	RAENet	ZPS	CE
	2021	Zhu et al.	41	DeepLab-V3+	ZPS	CE
	2022	Vengala et al.	42,43	TriNet	GSF	MSE + CE
		Zhang and Li	44	EESANet	GSF	Weighted CE
dDN	2020	Yan et al.	45	ResNet	ZPS	MSE

It is well-known that training datasets will affect network performance. Simulation is another convenient and important option for dataset generation, in addition to using real samples to obtain absolute phases. Thus, we will evaluate the effectiveness of the dataset generation methods as a preparation work in Sec. 3, and then compare different phase-unwrapping methods in different cases such as noise, discontinuity, and aliasing in Sec. 4.

In all these comparisons, the following three indices are used for accuracy estimation:

(i) RMSE: the traditional root mean squared error, including the mean of RMSE ($\mathrm{RMSE_m}$) and the standard deviation of RMSE ($\mathrm{RMSE_{sd}}$). $\mathrm{RMSE_{sd}}$ is used to indicate the stability of performance.

(ii) PFS: the proportion of failed samples. A failed sample is one for which there is at least one pixel with an absolute error greater than π. The congruence operation cannot correct such pixels.

(iii) PIP: the mean proportion of incorrect pixel for the failed samples. The incorrect pixel is one whose error is nonzero or greater than π.

2.5 Implementation of the Deep-Learning-Involved Methods

For dRG and dWC, the Res-UNet is selected from Ref. 72 (see Sec. S1 in the Supplemental Material), in which the inception module is introduced into the residual block.[24,47,51,78] For dDN, ResNet is obtained by removing the max-pooling and transposed-convolution layers from Res-UNet and changing the channel number of all middle layers to 64. The dWC method is regarded as a classification/segmentation problem of eight categories, including the wrap counts from 0 to 7.

We use the adaptive moment estimation based optimization to train all the networks. The batch size is 16 and the learning rate is 0.01 (85% drop per epoch if the learning rate is $>10^{-6}$). The epoch size is 100 for all the datasets. The MAE loss function is used for dRG and dDN, while the composite loss function is used for dWC, which can be expressed as

$$\text{Composite loss}(\boldsymbol{r}) = \frac{1}{M}\left\{-\sum_{m=1}^{M} k_G(\boldsymbol{r})\log[k(\boldsymbol{r})] + \sum_{m=1}^{M} |\psi(\boldsymbol{r}) - \psi_G(\boldsymbol{r})|\right\}, \tag{6}$$

where $k_G(\boldsymbol{r})$ is the GT wrap counts, $k(\boldsymbol{r})$ is the network-output wrap counts, $\psi_G(\boldsymbol{r})$ is the GT absolute phases, $\psi(\boldsymbol{r})$ is the absolute phases obtained from the network-output wrap counts and the network inputs by Eq. (1), and M is the amount of data.

All the networks are implemented by Pytorch 1.0 based on Python 3.6.1, which is performed on a PC with Core i7-8700K CPU, 16 GB of RAM, and NVIDIA GeForce GTX 1080Ti GPU.

3 Phase Dataset Generation Methods

Due to the fundamental importance of datasets in deep learning, we introduce the absolute phase generation methods (Sec. 3.1) to prepare the datasets for deep-learning-involved phase unwrapping (Sec. 3.2) and evaluate their richness (Sec. 3.3).

3.1 Absolute Phase Generation Methods

We review three absolute phase generation methods, all of which undergo two steps: generate a random phase and then linearly scale it into a range of $[0, h]$ in radians. For the comparison, here the value range of h is uniformly set from 10 to 40. In practical applications, the range of h can be appropriately enlarged as required. The first step makes these methods different and will be described below, while the second step is the same for all the methods.

3.1.1 Random matrix enlargement (RME)

The RME method first generates small square matrices with different sizes (randomly set from 2 × 2 to 8 × 8), different data distribution types (uniform or Gaussian), and then enlarges them into a size of 128 × 128 by different interpolation methods (nearest, bilinear, or bicubic interpolation).[24,28,32] As an example, shown in Fig. 7, an initial small matrix is interpolated and enlarged into a big matrix, which is then linearly mapped to absolute phase with a higher h. In RME, the continuity is guaranteed by interpolation, while the randomness is introduced by the parameter selection of the initial small matrices.

3.1.2 Gaussian functions superposition (GFS)

The GFS method generates a random number of Gaussian functions (from 1 to 20) with different means (randomly set from 2 to 127 for x and y directions), variances (randomly set from 100 to 1000) and amplitudes (randomly set from 0 to 1), and then superposes them by addition or subtraction.[29,35,38,39,42–44] As an example, shown in Fig. 8, three 128 × 128 Gaussian functions are superposed by addition or subtraction, which are then linearly mapped to absolute phase with a higher h. In GFS, the continuity is guaranteed by using Gaussian functions, while the randomness is introduced by variations of Gaussian parameters and random function superposition.

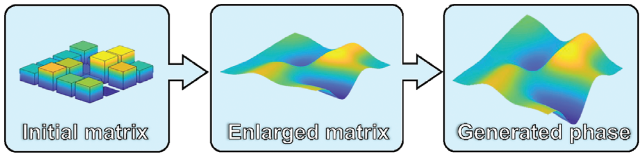


Fig. 7 An example of the RME method.

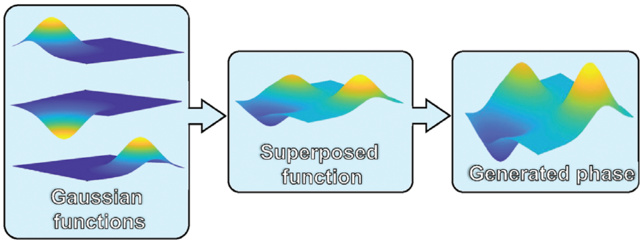


Fig. 8 An example of the GFS method.

3.1.3 Zernike polynomials superposition (ZPS)

The ZPS method generates a random number of matrices by the Zernike polynomials with different coefficients (randomly set from 0 to 1) of the first 30 orders (the first coefficient is set to zero), and then superposes these matrices by addition or subtraction.[36,37,40,41,45] The ZPS method is similar to the GFS method except that the Gaussian functions are replaced by Zernike polynomials in which the continuity is guaranteed by Zernike polynomials, while the random strengths of these polynomials introduce the randomness.

3.1.4 Real data reprocessing (RDR)

The RDR method obtains the absolute phases of real samples and then reprocesses them. The typical ways include: (i) using successful unwrapped results from a traditional phase unwrapping method[23,27,30,31,33]; (ii) using optical methods offering absolute phase, such as dual-wavelength (or even multiwavelength) digital holography method[79]; (iii) using phase-unwrapping-free methods such as the transport of intensity equation (TIE) method.[80]

3.2 Dataset Generation

Now that the absolute phase ψ has been generated, both the real and imaginary parts of the CAF are calculated as

$$R(x, y) = \cos[\psi(\boldsymbol{r})], \tag{7}$$

$$I(x, y) = \sin[\psi(\boldsymbol{r})], \tag{8}$$

from which, the wrapped phase φ is calculated as

$$\varphi(\boldsymbol{r}) = \arctan[I(\boldsymbol{r})/R(\boldsymbol{r})], \tag{9}$$

and the wrap count is calculated as

$$k(\boldsymbol{r}) = \text{round}\left[\frac{\psi(\boldsymbol{r}) - \varphi(\boldsymbol{r})}{2\pi}\right]. \tag{10}$$

We thus obtain the complete dataset as $D = \{\psi, \varphi, R, I, k\}$, from which $\{\varphi, \psi\}$, $\{\varphi, k\}$, and $\{R$ and I, R and $I\}$ are used as the input-GT pairs for dRG, dWC, and dDN, respectively.

Accordingly, we generate the respective datasets denoted as D_RME, D_GFS, and D_ZPS for three different absolute phase generation methods. The simulated datasets in this section do not contain any invalid pixels. As for the real dataset (the fourth dataset, D_RDR), after being obtained by the digital holography and the least squares (LS) phase unwrapping method,[11] the absolute phases of real samples are shifted so that the minimum value is equal to zero as the network GT. The wrapped phases are then calculated by Eqs. (7)–(9) as the network input. The real samples contain candle flames, pits of different arrangements, grooves of different shapes, and tables of different shapes. The sizes of the datasets are as follows: for D_RME, D_GFS, and D_ZPS, each contains 20,000 pairs for training and 2000 pairs for testing; D_RDR contains 421 pairs for testing. Given that a higher h corresponds to a more complex wrapped phase, we produce a larger proportion of the data with a higher h. Specifically, for the training part of these three datasets (D_RME, D_GFS, and D_ZPS), the h of 50% data is randomly selected from 10 to 30, 20% from 30 to 35, and 30% from 35 to 40; for the testing part of these three datasets, the h of all data is

Table 2 Summary of datasets. "—" indicates "not available."

Datasets	Size	Proportion of *h* from 10 to 30	Proportion of *h* from 30 to 35	Proportion of *h* from 35 to 40
Training part of D_RME	20,000	50%	20%	30%
Testing part of D_RME	2000	2/3	1/6	1/6
Training part of D_GSF	20,000	50%	20%	30%
Testing part of D_GSF	2000	2/3	1/6	1/6
Training part of D_ZPS	20,000	50%	20%	30%
Testing part of D_ZPS	2,000	2/3	1/6	1/6
D_RDR for testing	421	—	—	—

randomly selected from 10 to 40. The specific information of the datasets can be found in Table 2. Further, we compare the accuracy of the neural networks trained on D_RME and the dataset with a uniform distribution of *h* (called D_RME0) and find that the former is better (see Sec. S2 in the Supplemental Material).

3.3 Dataset Selection

3.3.1 Initial richness indication of different datasets

Shannon entropy,[81,82] as a measurement of the uncertainty of a random variable, can quantitatively characterize the amount of information contained in a dataset, which affects the generalization ability of the trained neural network. We compute the Shannon entropy of the absolute phases from the D_RME, D_GFS, and D_ZPS, as shown in Fig. 9. The means of the entropy from high to low are from RME (mean 4.41), GFS (mean 4.24), and ZPS (mean 4.03). This result gives an initial indication of the better information richness in D_RME and D_GFS.

3.3.2 Statistical gradient distribution of different datasets

The gradient (difference) distribution of absolute phases can also reflect the richness of the dataset. Thus, we first calculate the sum of the absolute values of the horizontal difference and vertical difference for each absolute phase in the dataset, and then obtain the statistical average gradient distribution (SAGD) of the dataset. Figure 10 shows the SAGD maps of different datasets, from which we can see that the SAGD values of the four corners of D_RME and D_GSF are low as shown by the red arrow, D_ZPS only has high SAGD value in the four corners as shown in the red circle, and the SAGD value of D_RDR is higher in the lower middle region as shown in the red circle.

3.3.3 Comparison with network's performance trained by different datasets

We train the same Res-UNet by D_RME, D_GFS, and D_ZPS. As a result, we have three trained networks named RME-Net, GFS-Net, and ZPS-Net, respectively. These three networks are then tested by all the testing datasets and the results are shown in Table 3. We observe that the networks trained with a single type of dataset (RME-Net, GFS-Net, and ZPS-Net) perform best on the test data from the same dataset, and slightly worse on those from other datasets including RDR. This is the so-called generalization problem. For generalization capability, RME-Net is similar to GFS-Net and better than ZPS-Net, which is consistent with the Shannon entropy. What's more, the congruence operation can significantly improve the accuracy of the networks (see Sec. S3 in the Supplemental Material).

To further analyze the error distribution of the neural network and explore its relationship with the SAGD of the dataset, we calculate the mean error map for each network on each testing dataset. As shown in Fig. 11, RME-Net generalizes well to all testing datasets, GSF-Net generalizes poorly to D_RME, and

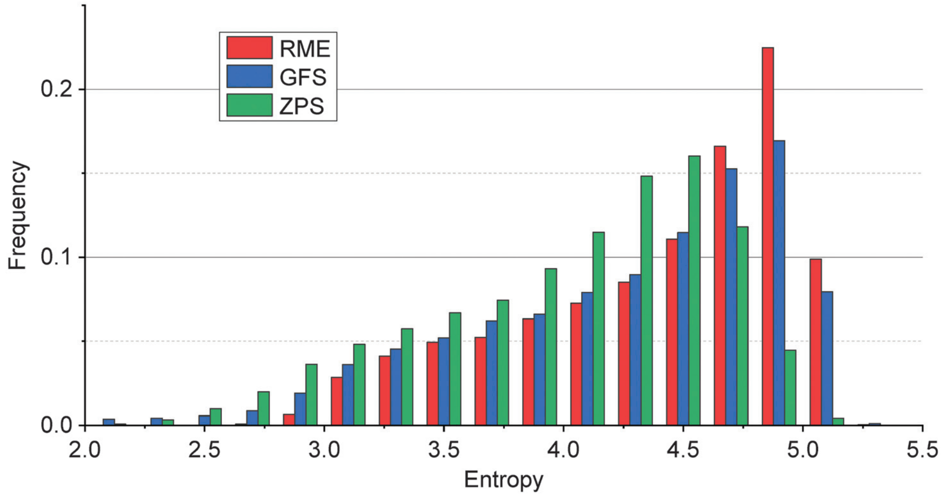


Fig. 9 Entropy histogram of absolute phases from the D_RME, D_GFS, and D_ZPS.

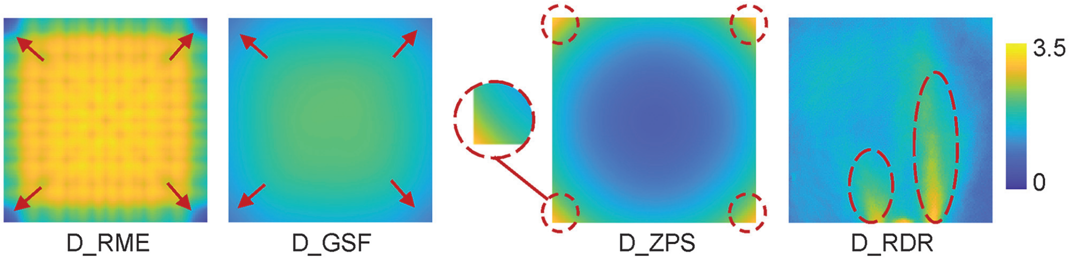


Fig. 10 SAGD maps of different datasets. Red arrows and circles indicate low and high SAGD values, respectively.

Table 3 RMSE_m, RMSE_sd, and PFS of phase unwrapping results of RME-Net, GFS-Net, and ZPS-Net.

		D_RME	D_GFS	D_ZPS	D_RDR
RMSE_m	RME-Net	0.0910	0.0982	0.1336	0.1103
	GSF-Net	0.2263	0.0985	0.1133	0.1184
	ZPS-Net	2.5148	0.4221	0.0821	0.8245
RMSE_sd	RME-Net	0.0507	0.1037	0.2320	0.1003
	GSF-Net	0.4571	0.0234	0.1077	0.1557
	ZPS-Net	2.8249	0.6252	0.0220	1.1405
PFS	RME-Net	0.0010	0.0085	0.1270	0.0594
	GSF-Net	0.1485	0.0020	0.0560	0.0333
	ZPS-Net	0.6525	0.4075	0.0010	0.4679

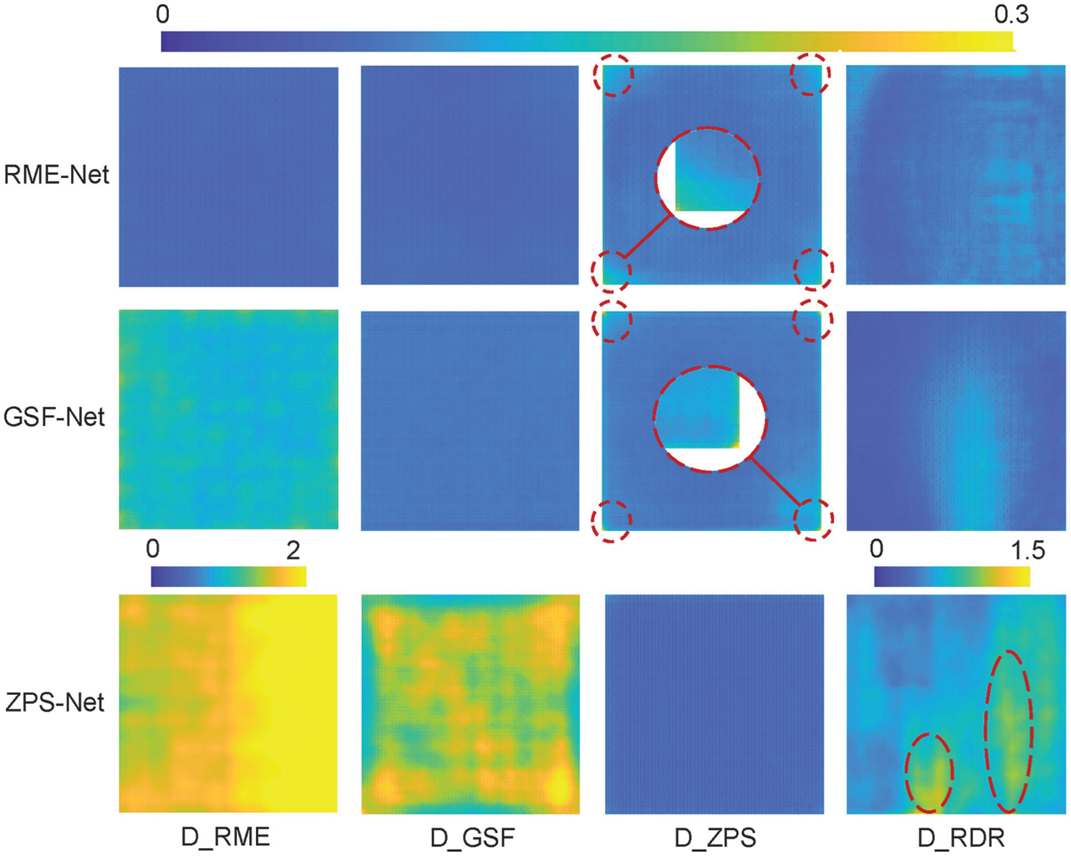


Fig. 11 Mean error maps for each network. Red circles indicate high mean error value.

ZPS-Net generalizes poorly to all non-ZPS testing datasets. This is consistent with the results in Table 3. Furthermore, since the SAGD value on the four corners of D_RME and D_GSF is low but the SAGD value on the four corners of D_ZPS is high, the mean error of RME-Net and GSF-Net on D_ZPS is higher on the four corners as indicated by the small red circles. Similarly, since D_ZPS has a low SAGD value in the regions where the SAGD value of D_RDR is high, the mean error of ZPS-Net on D_RDR is larger in the corresponding regions as indicated by the large red ellipses.

Therefore, the best performing RME method is selected and further improved to obtain a dataset with a uniform SAGD. Specifically, based on the RME method, we enlarge the small square matrices into a size of 160 × 160 and then intercept the central 128 × 128 part as the absolute phase, which can avoid a lower SAGD value at the four corners. Using this modified RME method, we generate a dataset of 20,000 pairs of data, called D_RME1, and use it to train the neural network to obtain REM1-Net. As shown in Fig. 12(a), the SAGD map of D_RME1 is more uniform than that of D_RME. Correspondingly, as shown in Fig. 12(b), compared with RME-Net, the mean error of RME1-Net decreases overall, and phase unwrapping can also be performed well for the four corners of the D_ZPS. Likewise, RMSE of RME1-Net is much lower than that of RME-Net (see Sec. S4 in the Supplemental Material).

For more visualization, in Fig. 13, we show the results of RME1-Net for the four testing datasets with the maximum, medians, and minimum RMSE_m. Each pixel of the RME1-Net results has a small error, most of which can be corrected by the congruence. The phase unwrapping task is indeed successfully fulfilled by a trained network.

Based on the above evaluation, we select the RME as the dataset generation method to do the following testing for different phase unwrapping methods. Furthermore, the congruence operation is always considered due to its high effectiveness. To facilitate the reader to further understand the deep-learning-involved phase unwrapping, we present a step-by-step example in Sec. S5 of the Supplemental Material, which includes dataset generation, neural network making, training, and testing.

4 Comparison of Deep-Learning-Involved Phase Unwrapping Methods

4.1 Dataset Preparation

We generate the absolute phase by the RME method. Then Eqs. (7) and (10) are used to get the ideal dataset $D = \{\varphi, \psi, k\}$. The pairs $\{\varphi, \psi\}$ and $\{\varphi, k\}$ are used as input-GT pairs for dRG and dWC, respectively.

To generate the noisy dataset, we add the noise of different degrees to ψ from which other necessary fields are computed. The complete noisy dataset is $D_n = \{\psi, \varphi_n, R, I, R_n, I_n, k\}$, where the subscript n is used to denote that the fields are noisy, and ψ, R, I, and k remain noiseless. The pairs $\{\varphi_n, \psi\}$, $\{\varphi_n, k\}$ and $\{R_n$ and I_n, R and $I\}$ are used as input-GT pairs for dRG, dWC, and dDN, respectively.

To generate the discontinuity-containing dataset, we place a rectangular area with a random size and phase value of π in ψ from which all other fields are generated in the same way described above. Accordingly, the complete dataset with a discontinuity is $D_d = \{\psi_d, \varphi_d, k_d\}$, where the subscript d is used to denote that the fields are discontinuous. The pairs $\{\varphi_d, \psi_d\}$ and $\{\varphi_d, k_d\}$ are used as input-GT pairs for dRG, and dWC, respectively. dDN is not involved in the comparison.

To generate the aliasing-containing dataset, we simply increase h and the size of the small square matrices to obtain the absolute phase with a steeper distribution. Accordingly, the complete dataset with aliasing is $D_a = \{\psi_a, \varphi_a, k_a\}$, where the subscript a is used to denote that the fields are aliasing. The pairs $\{\varphi_a, \psi_a\}$ and $\{\varphi_a, k_a\}$ are used as input-GT pairs for dRG, dWC, respectively. dDN is not involved in the comparison.

To generate the mixed dataset (containing all three of noise, discontinuity, and aliasing), first, we simply increase h and the size of the small square matrices to obtain the absolute phase

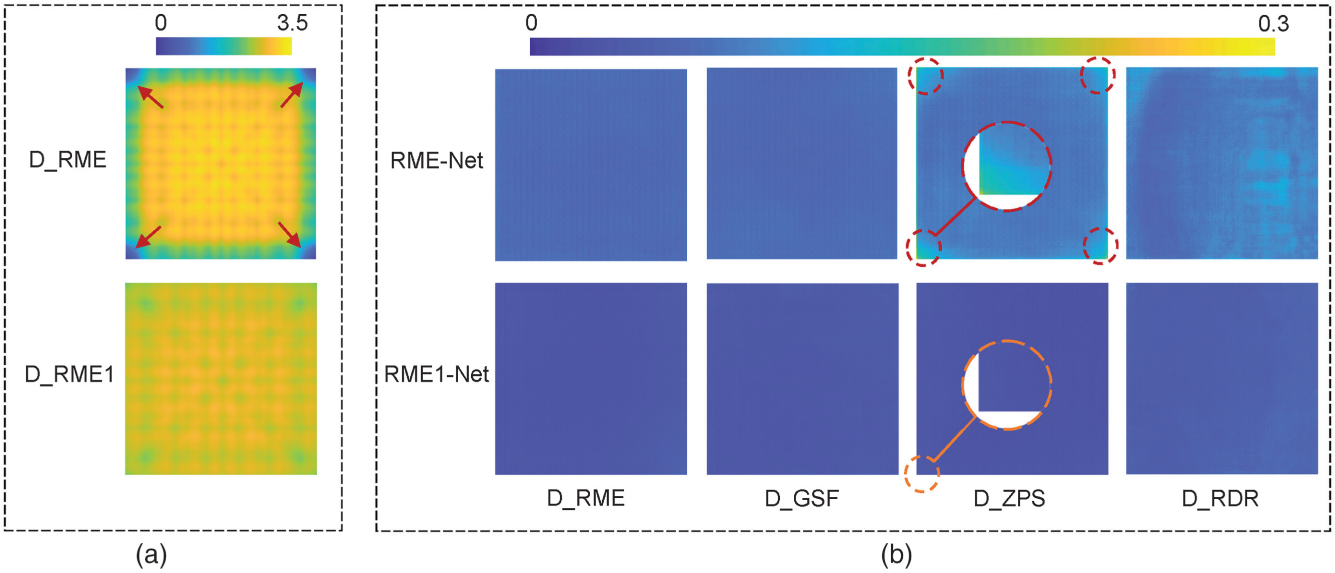


Fig. 12 (a) SAGD maps for D_RME and D_RME1, (b) mean error maps for RME-Net and RME1-Net. Red arrows indicate low SAGD value. Red circles indicate high mean error value and orange circles indicate the comparison part.

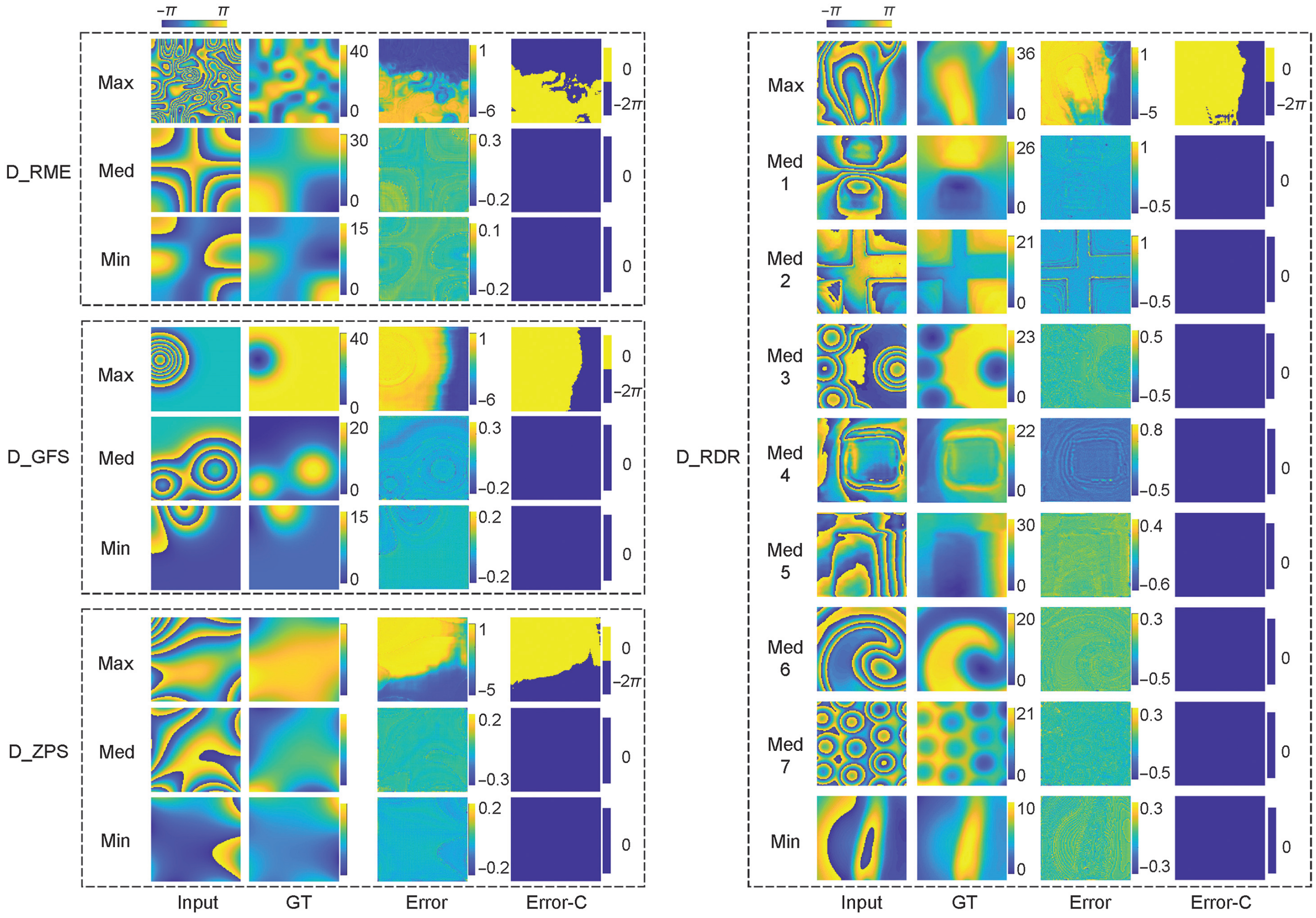


Fig. 13 Partial display of results from RME1-Net. "Max", "Med," and "Min" represent specific results with maximal, median, and minimal RMSE_m, respectively. "-C" represents the congruence results.

with a steeper distribution, then place a rectangular area with a random size and phase value of π in ψ, and then add the noise of different degrees to ψ. Accordingly, the mixed dataset is $D_m = \{\psi_m, \varphi_m, k_m\}$, where the subscript m is used to denote that the fields are mixed with all three of noise, discontinuity, and aliasing. The pairs $\{\varphi_m, \psi_m\}$ and $\{\varphi_m, k_m\}$ are used as input-GT pairs for dRG, dWC, respectively. dDN is not involved in the comparison.

For each noise-containing case, the Gaussian noise with a random standard deviation σ_n ranging from 0 to 1.8 is added to the pure ψ in which only the data with the wrapped phase $\mathrm{SNR} \geq -3$ is kept to eliminate invalid data caused by excessive noise. For each aliasing-free case, when generating datasets, there are 22,000 absolute phases with uniformly distributed h from 10 to 40 in which 20,000 for training and 2000 for testing. For each aliasing-containing case, we set the size of the initial random matrix from 8×8 to 12×12 and set h from 45 to 60. For each discontinuity-containing case, a square area with three random variables is simulated, including the x axis starting point (random value from 1 to 64), the y axis starting point (random value from 1 to 64) and the square size (random value from 20×20 to 50×50); in the square, the phase is set as 2π.

For clarity, Table 4 summarizes the different cases (column 1), the datasets (column 2), the networks (column 3), and the

Table 4 Summary of networks and corresponding datasets. The form of the dataset is {Input, GT}. The last letter of the network name is the case ("I" for ideal, "N" for noisy, "D" for discontinuous, "A" for aliasing, and "M" for mixed).

Cases	Datasets	Networks	Loss functions
Ideal case (Sec. 4.2)	$\{\varphi, \psi\}$	dRG-I	MAE
	$\{\varphi, k\}$	dWC-I	CE + MAE
Noisy case (Sec. 4.3)	$\{\varphi_n, \psi\}$	dRG-N	MAE
	$\{\varphi_n, k\}$	dWC-N	CE+MAE
	$\{R_n$ and I_n, R and $I\}$	dDN-N	MAE
Discontinuous case (Sec. 4.4)	$\{\varphi_d, \psi_d\}$	dRG-D	MAE
	$\{\varphi_d, k_d\}$	dWC-D	CE + MAE
Aliasing case (Sec. 4.5)	$\{\varphi_a, \psi_a\}$	dRG-A	MAE
	$\{\varphi_a, k_a\}$	dWC-A	CE + MAE
Mixed case (Sec. 4.6)	$\{\varphi_m, \psi_m\}$	dRG-M	MAE
	$\{\varphi_m, k_m\}$	dWC-M	CE + MAE

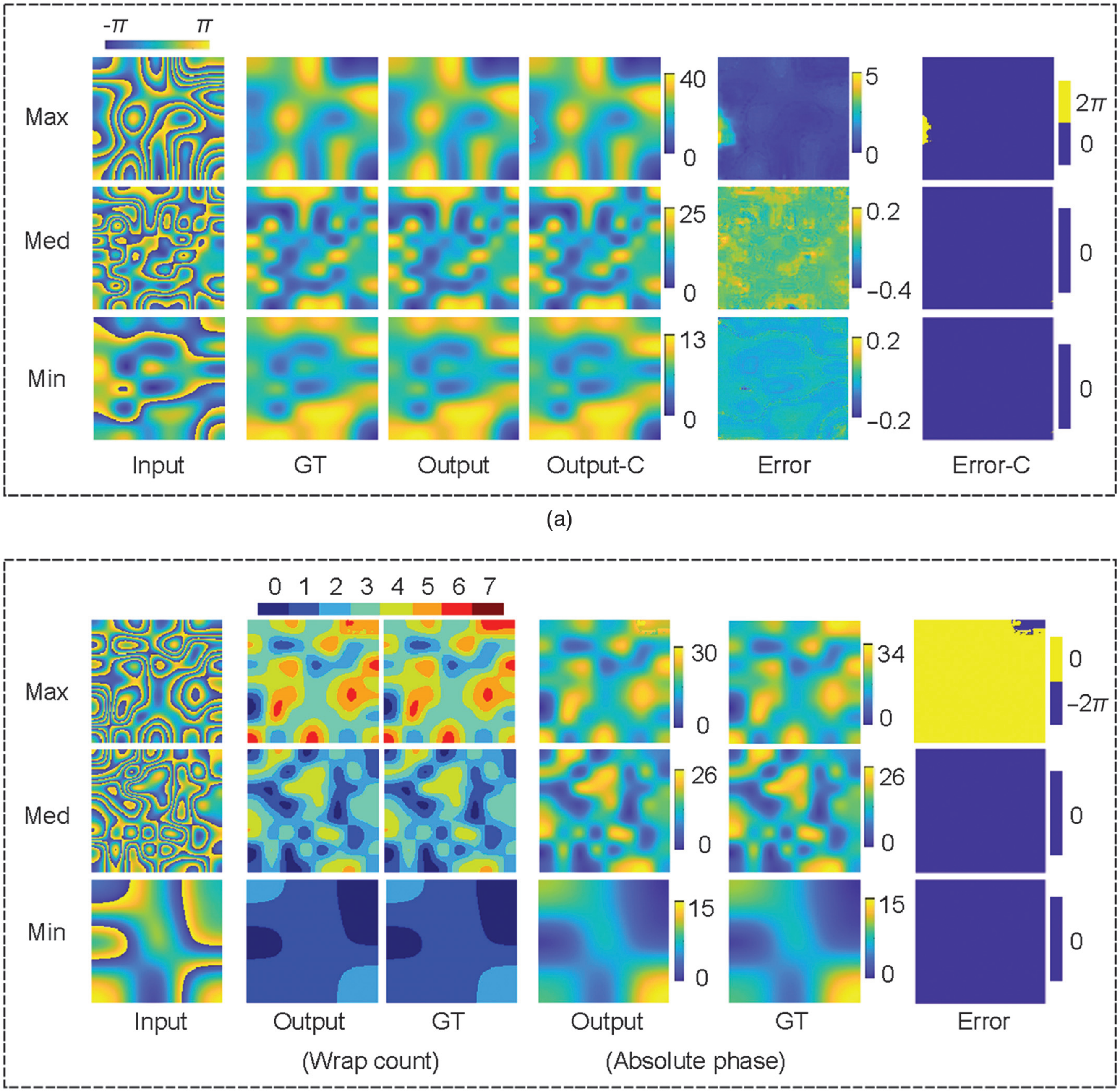
Table 5 $\mathrm{RMSE_m}$, $\mathrm{RMSE_{sd}}$, PFS, and PIP of the deep-learning-involved methods in the ideal case. "-C" represents the congruence results.

	dRG-I	dRG-I-C	dWC-I
$\mathrm{RMSE_m}$	0.0989	0.0005	0.0008
$\mathrm{RMSE_{sd}}$	0.0515	0.0157	0.0251
PFS	0.0015	0.0015	0.0025
PIP	0.0044	0.0044	0.0054

used loss functions (column 4). Due to the increase in h, for each aliasing-free case, the categories number of the network for dWC is increased from 8 to 10, including the wrap counts from 0 to 9. All information about neural network training can be found in Sec. 2.5.

4.2 Comparison in the Ideal Case

After training, we test the dRG-I and dWC-I. The accuracy evaluation of the networks is shown in Table 5 and Fig. 14. Both dRG with congruence and dWC provide high success



(a)

(b)

Fig. 14 Results for the (a) dRG-I and (b) dWC-I in the ideal case. "Max," "Med," and "Min" represent specific results with maximal, median, and minimal $\mathrm{RMSE_m}$, respectively. "-C" represents the congruence results.

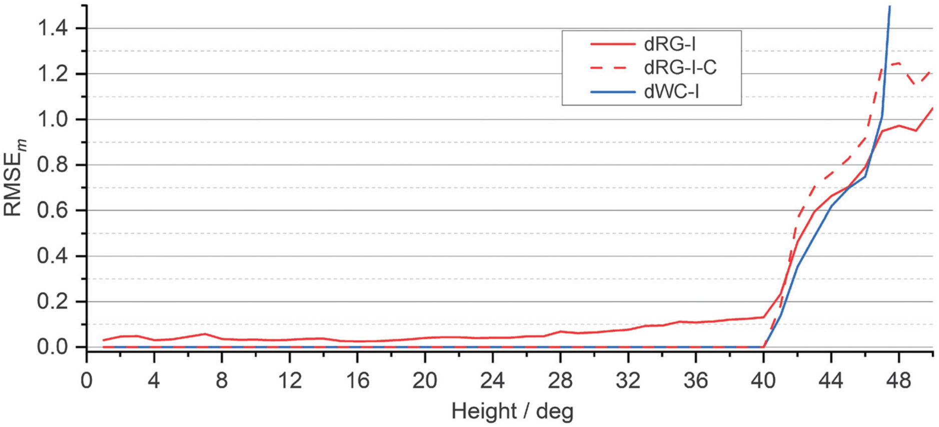


Fig. 15 RMSE$_\mathrm{m}$ of the deep-learning-involved methods for absolute phase in different heights.

rates, with PFS as low as 0.0015 and 0.0025. It should be noted that all three traditional methods, such as line-scanning, congruence of LS, quality guided (QG),[10] can get completely correct results, i.e., their PFS is 0. Therefore, on one hand, the deep-learning-involved methods do provide satisfactory results just by some training, and prompt us to further examine and compare them in other more challenging cases; on the other hand, the presence of failed results is annoying, although it is already the state-of-the-art results that we have tried to achieve by now. Ensuring that the neural network can correctly unwrap the wrapped phase with any shape is the focus of future research.

To test the accuracy and adaptation of the neural network to samples with different heights, a testing dataset with h from 1 to 50 is generated by RME to test dRG-I (dRG-I-C) and dWC-I, whose RMSE$_\mathrm{m}$ is shown in Fig. 15. Note that the networks can successfully complete the phase unwrapping of samples with $h < 40$, but performance degradation occurs for samples with $h > 40$. Further, we train other neural networks by the datasets with h in the range of [10, 80] and perform similar tests, and the height adaptive range of the neural network to h is also increased accordingly (see Sec. S6 in the Supplemental Material), which means that in practical applications, the height range of the training dataset should be appropriately expanded, especially for the upper limit of h.

4.3 Comparison in the Noisy Case

After training, 2000 pairs of samples in the testing datasets are used to test the corresponding networks, whose accuracy evaluation is shown in Table 6 and Fig. 16. We have the following observations:

i. After congruence, RMSE$_\mathrm{m}$ of dRG and dDN becomes almost the same as that of dWC, which proves the good effect of congruences.

ii. For dRG without congruence, the RMSE$_\mathrm{m}$ is only slightly higher than that of the ideal case, which is because that dRG implicitly includes denoising when it is trained by the noisy wrapped phase as input and the pure absolute phase as GT. This result shows that the implicit denoising in dRG is almost as effective as dDN.

iii. For dWC, the neural network only predicts the wrap count, with which the noisy wrapped phase is unwrapped. Thus, the unwrapped phase is also noisy. The noise will also be included if the pure absolute phase (GT) is used to calculate the unwrapping error. Thus, we use the noisy absolute phase (GT1) to calculate the unwrapping error, as shown in the red enlarged boxes in Fig. 16(b). This consideration is also incorporated in Table 6 and Fig. 17.

Table 6 RMSE$_\mathrm{m}$, RMSE$_\mathrm{sd}$, PFS, and PIP of the deep-learning-involved methods in the noisy case. "GT" represents the pure GT (pure absolute phase), while "GT1" represents the noisy GT (noisy absolute phase). "-C" represents the congruence results.

	dRG-N (GT)	dRG-N-C (GT1)	dWC-N (GT1)	dDN-N (GT)	dDN-N-C (GT1)
RMSE$_\mathrm{m}$	0.1367	0.0285	0.0435	0.0883	0.0229
RMSE$_\mathrm{sd}$	0.1154	0.1148	0.1197	0.2915	0.3056
PFS	0.2525	0.2525	0.2840	0.1976	0.1976
PIP	0.0013	0.0013	0.0014	0.0108	0.0088

iv. For dDN without congruence, the RMSE$_\mathrm{m}$ is the lowest, since dDN is specifically designed for denoising. However, if only one pixel is not well filtered, it could result in error propagation in the subsequent line-scanning method, as shown in the red enlarged boxes and arrows in Fig. 16(c). This is also the reason why the PFS of dDN-N-C is smaller than that of dRG-N-C, but the PIP of dDN-N-C is larger than that of dRG-N-C. With interest, we also train a neural network to denoise the wrapped phase directly, whose result is far worse than dDN (the RMSE$_\mathrm{m}$ is 26 times higher). It thus is not recommended because the denoised wrapped phase generally has minor errors where the wrapped fringe jumps, resulting in the error propagation in the subsequent line-scanning method (see Sec. S7 in the Supplemental Material).

To test the performance under different noise levels, we generate another noise-increasing dataset by adding Gaussian noise to five pure absolute phases in which the SNR of the wrapped phase gradually decreases from 20 to −3 in 0.5 intervals. As a comparison, the traditional line-scanning, LS, QG, and window-Fourier-transform-preceded quality guided (WFT-QG) methods are also tested.[7,13,14] The WFT parameters are selected from a series of values in a range: $\sigma_x = \sigma_y = [3:2:7]$ for window sizes, $\xi_{xl} = \xi_{yl} = [-0.2: -0.1: -3.0]$ for low-frequency bounds, $\xi_{xh} = \xi_{yh} = [0.2:0.2:3.0]$ for high-frequency

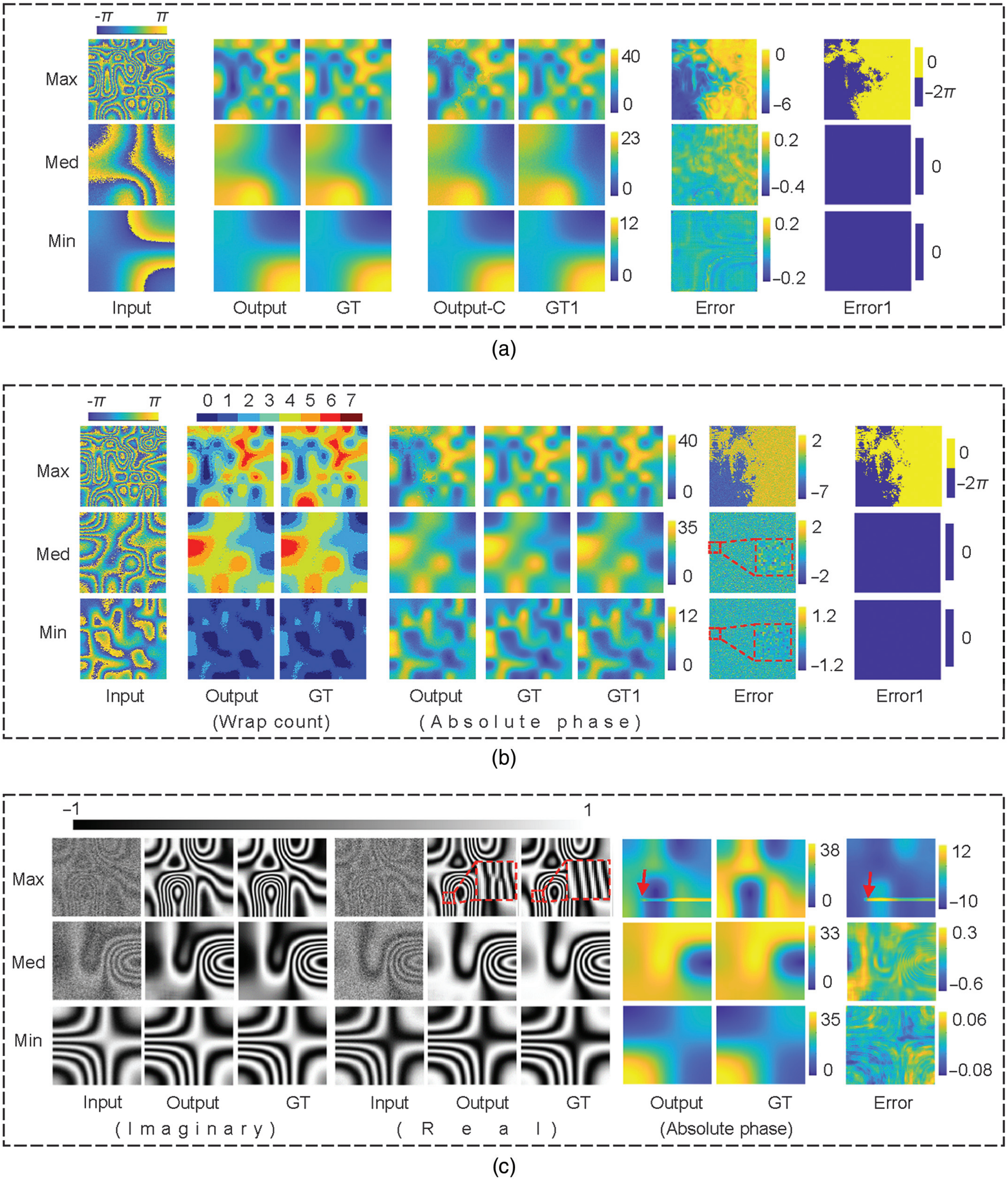


Fig. 16 Results for (a) dRG-N, (b) dWC-N, and (c) dDN-N in the noisy case. "GT" represents the pure GT (pure absolute phase), while "GT1" represents the noisy GT (noisy absolute phase). "Max," "Med," and "Min" represent specific results with maximal, median, and minimal $\mathrm{RMSE_m}$, respectively. "-C" represents the congruence results.

bounds and thr $= [0.2:0.2:6.0]$ for threshold, where $[x:y:z]$ represents an arithmetic sequence with interval y from x to z; $\xi_{xi} = 1/\sigma_x$ and $\xi_{yi} = 1/\sigma_y$ are accordingly determined for the increasement of frequency.[83] This setting produces many parameter combinations, i.e., for each wrapped phase, WFT is applied multiple times with different parameters, and the one with the smallest $\mathrm{RMSE_m}$ is chosen as the final result. The average $\mathrm{RMSE_m}$ of these five groups are plotted at different noise levels

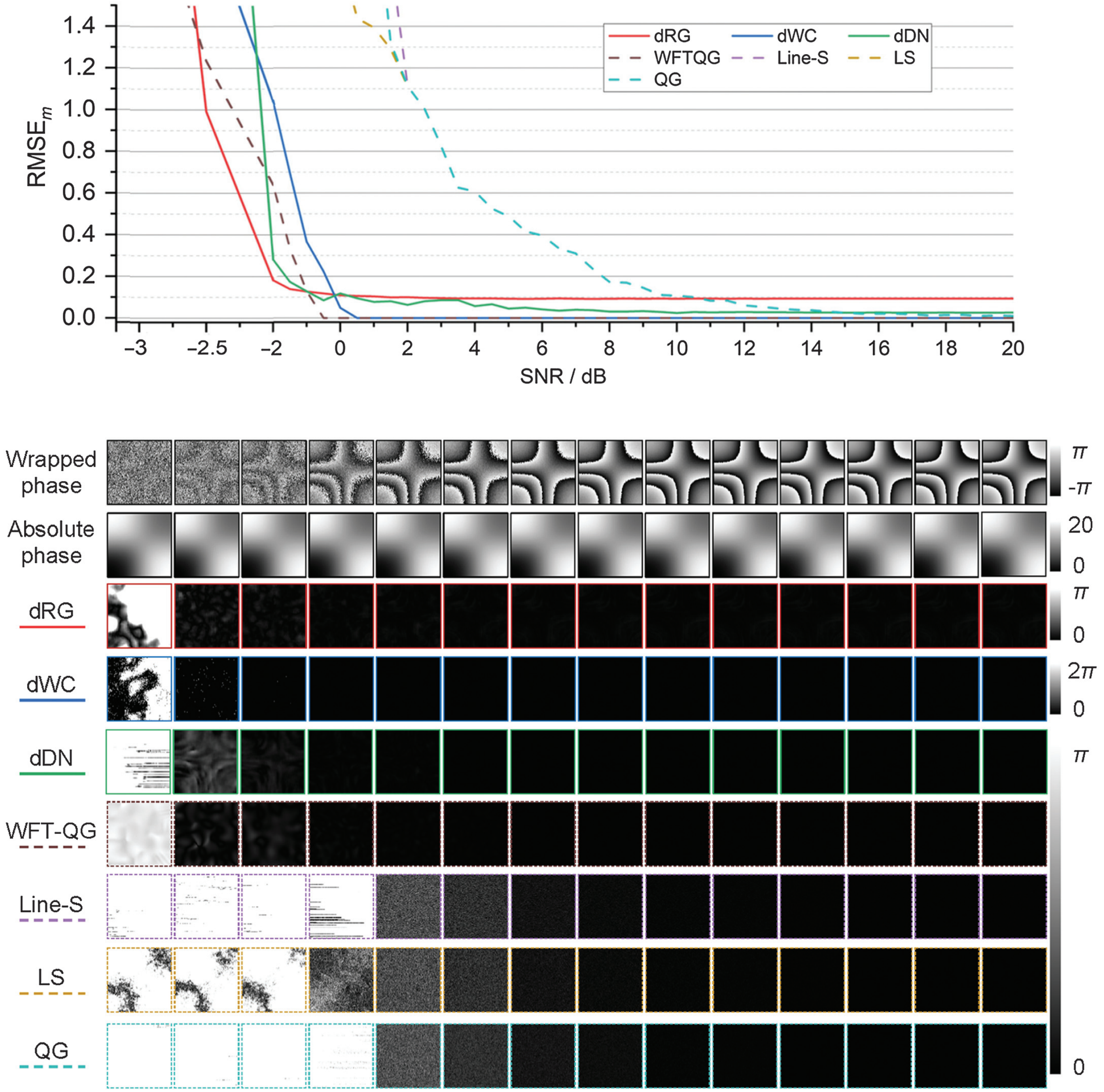


Fig. 17 Results in different noise levels. Solid and dashed lines represent the deep-learning-involved and traditional methods, respectively.

in Fig. 17. For intuition, we select an example and show the wrapped phase, the absolute phase, the absolute error maps of all the methods in the lower part of Fig. 17. The SNR of each column wrapping phase corresponds to the x axis. We have the following observations:

i. All methods are satisfactory when the SNR of the wrapped phase is >10.

ii. The errors of line-scanning, LS, QG start to increase rapidly after the SNR of the wrapped phase is <10.

iii. The performance of dWC and WFT-QG begins to degrade rapidly after the SNR of the wrapped phase is <0.

iv. The dRG and dDN do not show a significant performance degradation until the SNR in the wrapping phase is ≤ -2.

v. When the SNR of the wrapped phase is >0, the RMSE$_m$ of dWC and WFT-QG is the lowest, and the RMSE$_m$ of dRG is the highest. It should be noted that the congruence can significantly reduce the RMSE$_m$ of dRG and dDN (see Sec. S8 in the Supplemental Material).

Deep-learning-involved methods and WFT-QG can cope with the noise level with the wrapped phase SNR as low as 0 or even lower, but WFT-QG needs to be premised on suitable manually found hyperparameters that usually require several attempts. Other traditional methods can only maintain high accuracy in low to medium noise where the SNR of the wrapped phase is >10. It should be noted that dDN-N can also complete the phase unwrapping of ideal samples and obtain an accuracy comparable with dRG-I.

4.4 Comparison in the Discontinuous Case

After training, we use the testing datasets to test the corresponding networks and traditional methods, whose accuracy evaluation is shown in Table 7 and Fig. 18. All the indices in

Table 7 RMSE_m, $\mathrm{RMSE}_\mathrm{sd}$, PFS, and PIP of the deep-learning-involved and traditional methods in the discontinuous case. "-C" represents the congruence results.

	dRG-I	dRG-D	dRG-D-C	dWC-I	dWC-D	Line-scanning	LS	QG
RMSE_m	2.0230	0.1230	0.0261	1.2209	0.0219	3.8054	1.3655	2.4204
$\mathrm{RMSE}_\mathrm{sd}$	1.7817	0.1636	0.1827	1.3777	0.1543	3.7172	1.0408	2.5014
PFS	0.8120	0.0770	0.0770	0.7385	0.0785	0.9405	0.7120	0.8565
PIP	0.2407	0.0112	0.0112	0.1128	0.0077	0.4400	0.1073	0.2789

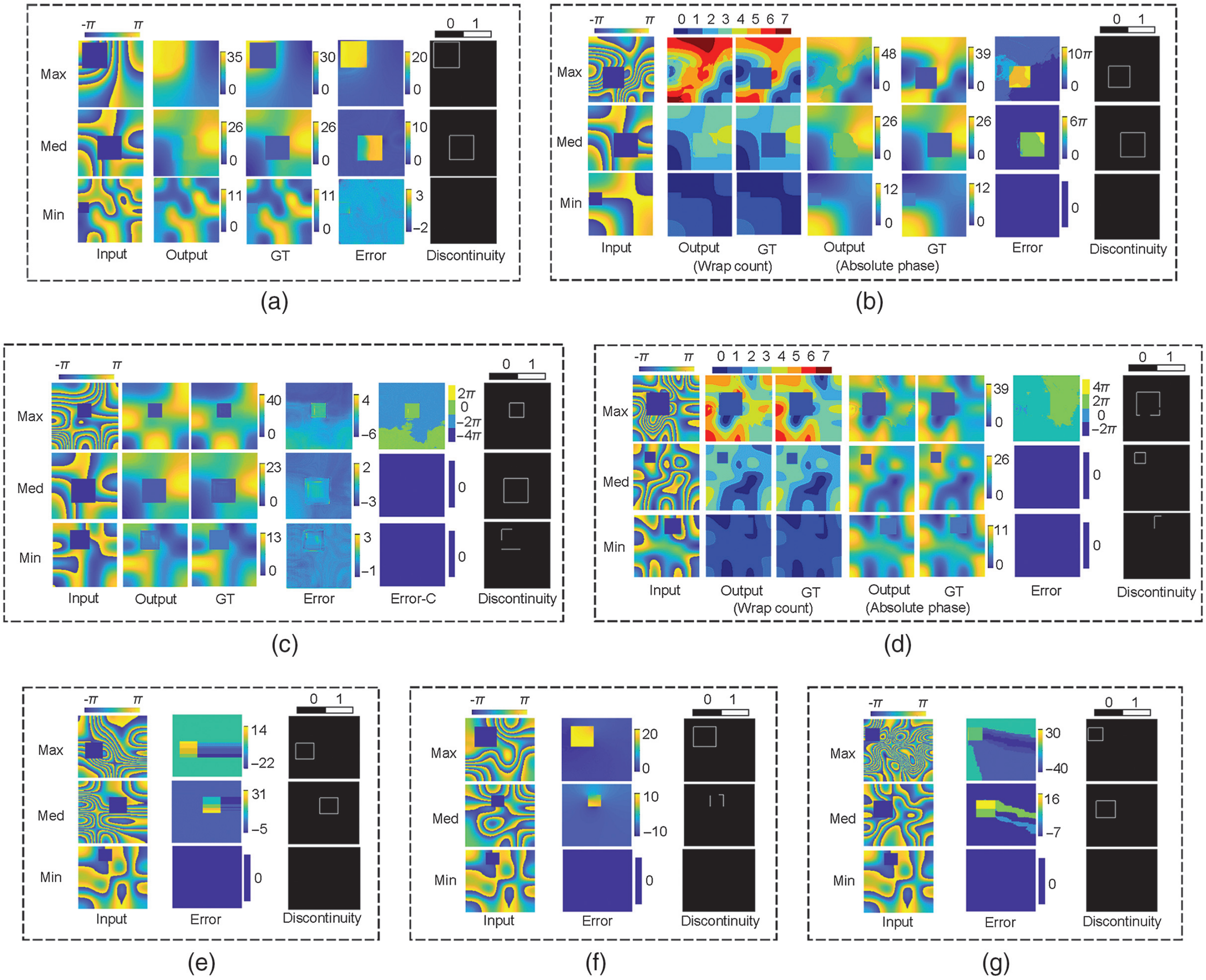


Fig. 18 Results for (a) dRG-I, (b) dWC-I, (c) dRG-D, (d) dWC-D, (e) line-scanning, (f) LS, and (g) QG methods in the discontinuous case. "Max," "Med," and "Min" represent specific results with maximal, median, and minimal RMSE_m, respectively. "-C" represents the congruence results. The last columns of each result are discontinuous maps, where 1 (white) represents the position of the discontinuous pixels.

Table 7 are calculated from pixels outside the square area. We have the following observations:

i. Overall, deep learning methods (dRG-D and dWC-D) significantly outperform traditional methods for phase unwrapping of discontinuous samples.

ii. As shown in Figs. 18(a) and 18(b), the pixels in the square area for the dRG-I and dWC-I are not unwrapped correctly, but most pixels outside the area are unaffected. Further, as shown in Fig. 18(c) and 18(d), after being trained by the discontinuity-enhanced dataset, the dRG-D and dWC-D can correctly unwrap each area.

iii. As shown in Figs. 18(e)–18(g), the three traditional methods get completely correct phase unwrapping results in the last row, where the square area does not introduce discontinuity. However, once the discontinuity appears in the first two rows, they all produce serious error propagation. Note that for LS and QG, this error propagation can be avoided by shielding square areas with a mask prior.

4.5 Comparison in the Aliasing Case

After training, we test the dRG-A, dWC-A, line-scanning, LS, and QG methods, whose accuracy evaluation is shown in Table 8 and Fig. 19. We have the following observations:

i. Overall, deep learning methods significantly outperform traditional methods for phase unwrapping of aliasing samples.

ii. dRG and dWC do not have large errors due to aliasing, but their performance is slightly degraded compared with that in the ideal case.

iii. As shown in Figs. 19(c)–19(e), the three traditional methods get completely correct phase unwrapping results in the last row, where aliasing points do not appear in the wrapped phase. However, once aliasing points appear in the first two rows, for the line-scanning and QG methods, the aliasing-caused error propagates on the integration path, resulting in the worst results. Even for the LS method, the phase unwrapping results near the aliasing pixels are also greatly affected.

Table 8 $\mathrm{RMSE_m}$, $\mathrm{RMSE_{sd}}$, PFS, and PIP of the deep-learning-involved and traditional methods in the aliasing case. "-C" represents the congruence results.

	dRG-A	dRG-A-C	dWC-A	Line-scanning	LS	QG
$\mathrm{RMSE_m}$	0.1958	0.0078	0.0107	40.5128	6.7199	39.8846
$\mathrm{RMSE_{sd}}$	0.1390	0.1503	0.1612	21.0695	3.1294	23.0389
PFS	0.0075	0.0075	0.0120	0.9820	0.9895	0.9895
PIP	0.0765	0.0765	0.0467	0.9102	0.5705	0.8369

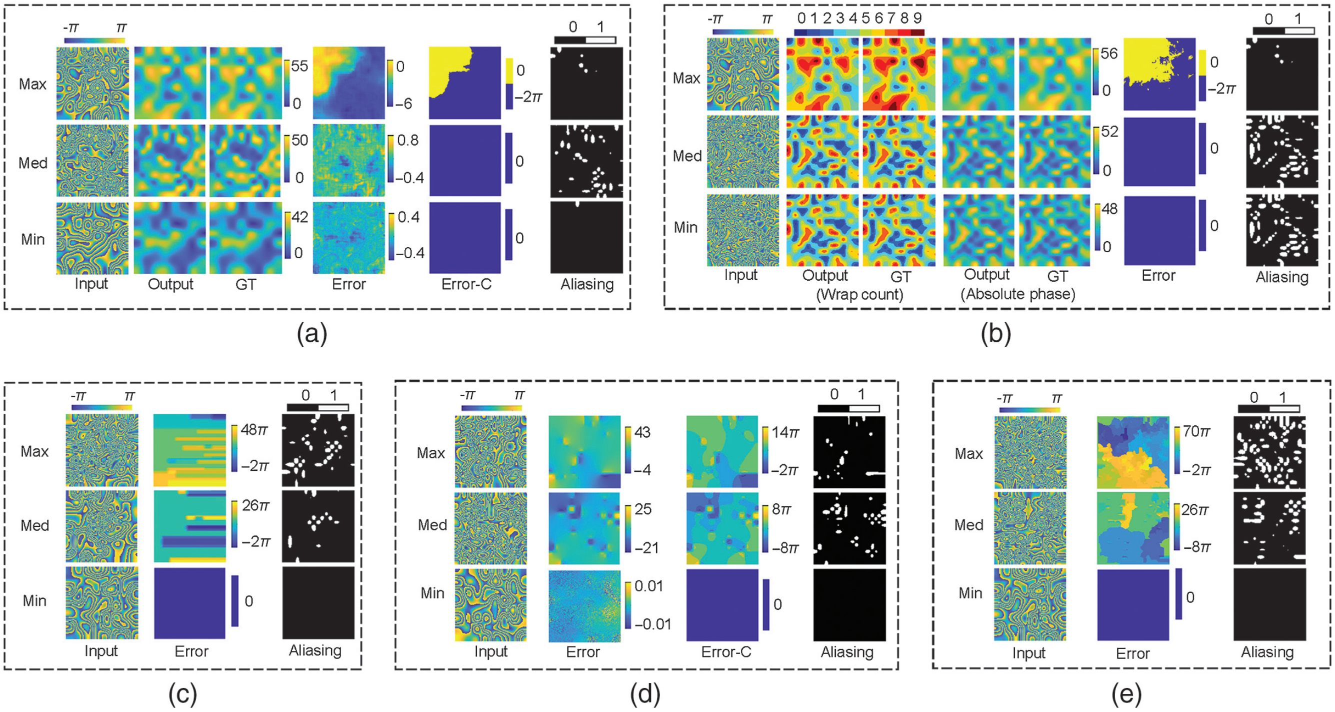


Fig. 19 Results for (a) dRG-A, (b) dWC-A, (c) line-scanning, (d) LS, and (e) QG methods in the aliasing case. "Max," "Med," and "Min" represent specific results with maximal, median, and minimal $\mathrm{RMSE_m}$, respectively. "-C" represents the congruence results. The last columns of each result are aliasing maps, where 1 (white) represents the position of the aliasing pixels.

4.6 Comparison in the Mixed Case

After training, we test the dRG-M, dWC-M, line-scanning, LS and QG methods, whose accuracy evaluation is shown in Table 9 and Fig. 20. We have the following observations:

i. Overall, deep learning methods significantly outperform traditional methods.

ii. Even if noise, discontinuity, and aliasing are present at the same time, dRG and dWC still maintain high accuracy. Moreover, the dRG after congruence is slightly more accurate than the dWC.

iii. All three traditional methods fail, whose PFS is 1, because of the presence of discontinuous or aliasing points in the wrapped phase of all samples. In contrast, the LS results have a better phase, but their $\mathrm{RMSE_m}$ is still about 49 times higher than that of dWC.

4.7 Performance Summary

Overall, each pixel of the unwrapped phase obtained by dRG and LS has a small error, which can be eliminated by the congruence operation, while the error of the unwrapped phase obtained by dWC, dDN, line-scanning, and QG is either 0 or integer multiples of 2π.

For the ideal case, all traditional methods (line-scanning, LS, and QG) can achieve perfect results, while deep-learning-involved methods (dRG, dWC, and dDN) cannot guarantee perfect completion of phase unwrapping of any shape due to their noninfinite generalization ability. The line-scanning method is most recommended due to its lowest computational cost.

With the introduction of invalid points (noise, discontinuities, and aliasing), the accuracy of deep-learning-involved methods

Table 9 $\mathrm{RMSE_m}$, $\mathrm{RMSE_{sd}}$, PFS, and PIP of the deep-learning-involved and traditional methods in the mixed case. "-C" represents the congruence results.

	dRG-M	dRG-M-C	dWC-M	Line-scanning	LS	QG
$\mathrm{RMSE_m}$	0.2362	0.1266	0.2206	38.4389	10.8350	39.4653
$\mathrm{RMSE_{sd}}$	0.3101	0.3790	0.4618	21.0695	3.6269	18.1084
PFS	0.3740	0.3740	0.4810	1.0000	1.0000	1.0000
PIP	0.0106	0.0106	0.0107	0.9569	0.7600	0.9107

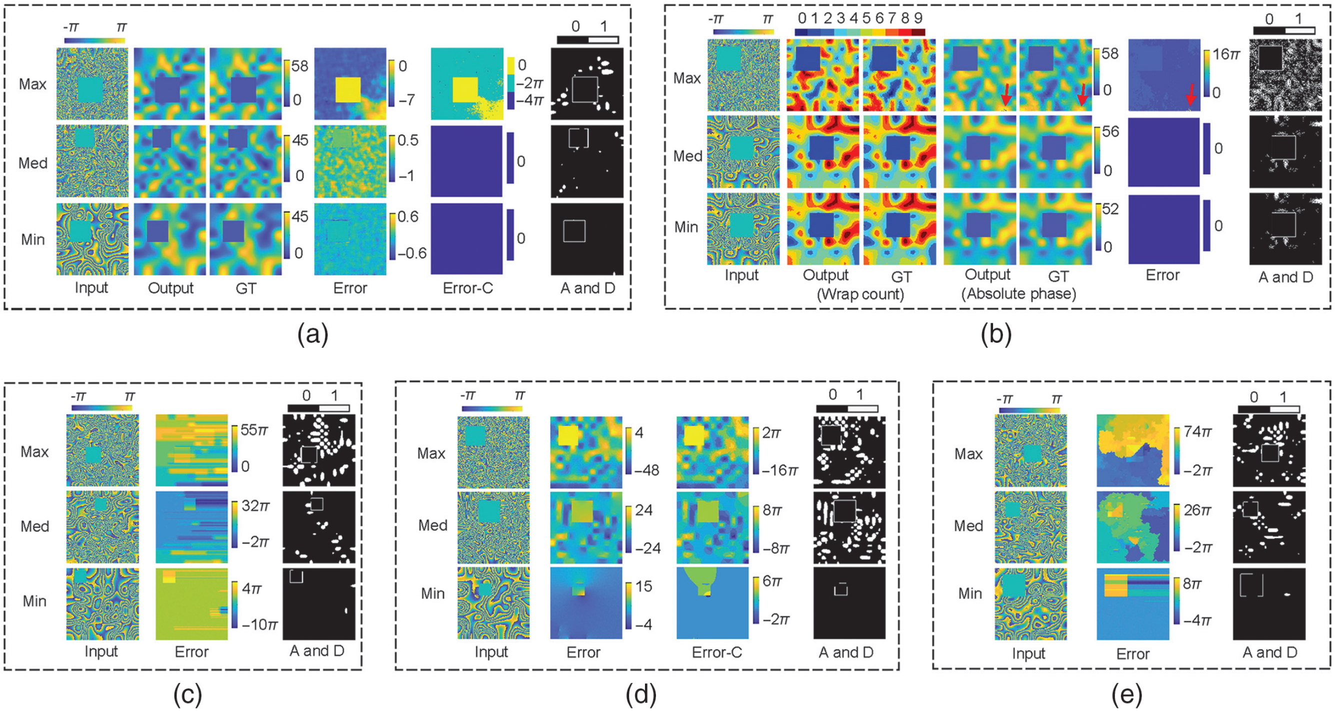


Fig. 20 Results for (a) dRG-M, (b) dWC-M, (c) line-scanning, (d) LS, and (e) QG methods in the mixed case. "Max," "Med," and "Min" represent specific results with maximal, median, and minimal $\mathrm{RMSE_m}$, respectively. "–C" represents the congruence results. The last columns of each result are aliasing or discontinuous maps (called "A and D"), where 1 (white) represents the position of the aliasing or discontinuous pixels.

Table 10 Performance statistics in the ideal, noisy, discontinuous, and aliasing cases. "✓" represents "capable." "✓✓" represents "best and recommended." "×" represents "incapable." "—" indicates "not applicable."

Cases	dRG	dWC	dDN	Line-scanning	LS	QG	WFT-QG
Ideal	✓	✓	✓	✓✓	✓	✓	—
Slight noise	✓	✓	✓	✓	✓	✓	✓
Moderate noise	✓✓	✓✓	✓	×	×	×	✓✓
Severe noise	✓✓	✓✓	✓	×	×	×	✓✓
Discontinuity	✓✓	✓✓	—	×	×	×	—
Aliasing	✓✓	✓✓	—	×	×	×	—
Mixed	✓✓	✓✓	—	×	×	×	—

has a corresponding slight drop but is basically lower than that of traditional methods.

For slight noise (SNR > 10), all methods can achieve satisfactory results. For moderate and heavy noise (SNR is <10 or even 0), the errors of line-scanning, LS, and QG methods become intolerable, and some errors of the neural network result of dDN will further cause the error propagation of the subsequent line-scanning method. Therefore, dRG, dWC, and WFT-QG are recommended. It should be noted that the unwrapped phase obtained by dWC is noise-containing, while dRG and WFT-QG can first obtain the pure unwrapped phase and, if necessary, the unwrapped phase with noise can be obtained by the congruence operation. The precondition for WFT-QG to obtain the best result is suitable manually found hyperparameters which usually require several attempts.

For discontinuous, aliasing, and even mixed cases, dDN and all traditional methods become powerless due to the Itoh condition, while dRG and dWC can still achieve satisfactory results after targeted training. Therefore, dRG and dWC are recommended as the only options.

To sum up, in Table 10, we present the performance of different methods in the different cases mentioned above.

5 Conclusion and Outlook

We have reviewed and compared the phase datasets generation methods (RME, GFS, and ZPS) and deep-learning-involved phase unwrapping methods (dRG, dWC, and dDN). When comparing deep-learning-involved phase unwrapping methods, the traditional methods were also tested for reference. For the dataset selection, the modified RME method is more recommended. The traditional phase unwrapping methods are more reliable and efficient in ideal, slightly, and moderately noisy cases, except WFT-QG, which can get satisfactory accuracy in all noisy cases. The deep-learning-involved methods are more suitable for situations when most of the traditional methods are powerless, such as severe noisy, discontinuous, and aliasing cases.

We aim to provide a relatively uniform and fair condition for different dataset generation methods and phase unwrapping methods. In actual use, the parameter ranges of the three dataset generation methods (such as the initial matrix size in the RME method, the number of Gaussian functions in the GFS method, and the order number of the Zernike coefficient in the ZPS method) can be further expanded for more richness. For the fairness of comparisons and the efficiency of network training, we selected 20,000 as the number of training samples. More samples in practical applications will surely bring further improvement in network performance. In addition to the dataset generation methods mentioned above, the GAN is expected to generate more data based on a small amount of reliable data, thereby further increasing the richness of the dataset.[84] Once the training is complete, the size of the neural network input is fixed to the size of the training data. In practical applications, the size of the data to be unwrapped is generally different from that of the neural network input. To dynamically adapt the resolution, we propose to first divide the entire data into multiple overlapping subregions of fixed size, then do phase unwrapping for each subregion by the neural network, and finally use the stitching algorithm to obtain the entire absolute phase.[40,85,86]

The neural networks used in the above comparations are all based on Res-UNet. Some other types of network structures (such as Bayesian network,[87] dynamic convolution,[88] and attention UNet[89]) may further improve the accuracy of phase unwrapping.

All the deep-learning-involved methods mentioned in this paper rely on dataset supervision to learn the mapping relationship from input to GT. The wrapped phase can be unwrapped by passing through the trained neural network only once, but its generalization ability for samples with different shapes is not infinite.

Different types of objects contain different shape features in their phase distributions. For example, as a common object of interferometry, the phase distribution of the lens surface has a style of slow fluctuation and low-frequency information. As a

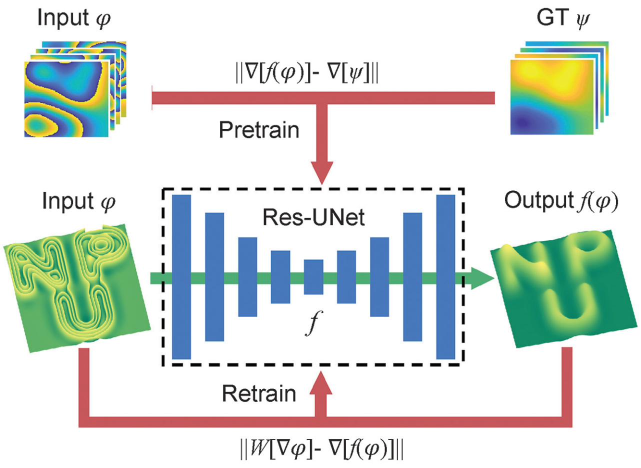


Fig. 21 Schematic diagram of pretraining and retraining.

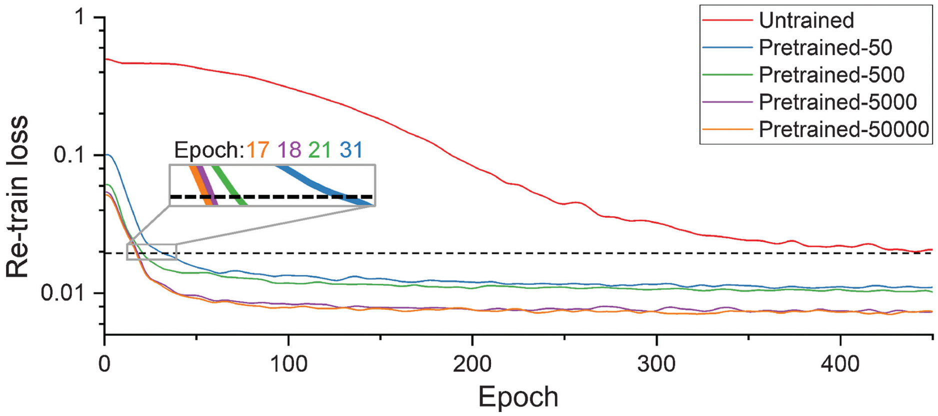


Fig. 22 Loss plot of pretrained and initialized networks.

common object of InSAR, the phase distribution of mountainous terrain has a style of obvious tortuous and high-frequency information. As a common object of holographic microscopy, the phase distribution of cells usually has a style that considers both high-frequency and low-frequency information. Therefore, we believe that a promising way to solve the problem of generalization ability is transfer learning.[90] The transfer learning way is, first, pretraining a standard neural network with a simulated dataset; then, for a specific target object, using the real dataset of this type of object to fine-tune the neural network; finally, unwrapping the wrapped phase through the fine-tuned neural network at one time.

But what if there is no real dataset of the target object? Different from the dataset-supervision method, inspired by deep image priori,[91] Yang et al.[92] used the Itoh condition to guide the convergence of the neural network as a physical-model supervision method, without GT and with stronger generalization ability for samples with different shapes. However, the initialized network does not include any mapping relationship from the wrapped phase to the absolute phase, leading to a large number of iterations for each phase unwrapping. Similar to transfer learning, we propose to combine the dataset supervision with physical-model supervision, which can significantly reduce the number of iterations. That is, pretraining the neural network by the simulation-generated dataset, and then fewer iterations are required for the physical-model supervision method.

Here, we show the preliminary results of this idea. To make the input and the neural network related, we change the input of the neural network in Ref. 92 from a random vector to a wrapped phase. We use the method in Sec. 4.1 to generate a training dataset containing 50,000 pairs of data, $D = \{\varphi, \psi\}$, and then randomly take out 5000, 500, and 50 pairs as the other three training datasets.

As shown in Fig. 21, by the pretrain loss function,

$$\text{Pretrain loss} = \|\nabla[f(\varphi)] - \nabla[\psi]\|, \tag{11}$$

we use the four training datasets to pretrain Res-UNet. Then, the four pretrained Res-UNets are used to unwrap a wrapped phase by the physical-model supervision with the following retrain loss function:

$$\text{Retrain loss} = \|W[\nabla\varphi] - \nabla[f(\varphi)]\|. \tag{12}$$

For comparison, we also use the physical-model supervision to train an initialized Res-UNet without pretraining.

From the retrain loss plot of the five networks shown in Fig. 22, we can find that the convergence speed of the pretrained networks is significantly faster than the initialized one without pretraining. Specifically, the initialized network without pretraining requires 500 epochs to converge, while the pretrained neural networks can converge in <100 epochs with higher precision. It is more interesting that with only one pretraining by the dataset of 50 pairs, the epoch required to achieve the same accuracy can be reduced from 500 to 32; further, by increasing the pretraining datasets of 50 pairs to 50,000 pairs, the required epoch also reduces from 32 to 17.

This idea of pretraining with dataset supervision and then retraining with physical-model supervision also has great potential in other fields, such as phase imaging,[93] coherent diffractive imaging,[94] and holographic reconstruction.[95]

Acknowledgments

National Natural Science Foundation of China (61927810, 62075183); NSAF Joint Fund (U1730137); Fundamental Research Funds for the Central Universities (3102019ghxm018).

References

1. L. Aiello et al., "Green's formulation for robust phase unwrapping in digital holography," *Opt. Lasers Eng.* **45**(6), 750–755 (2007).
2. M. Jenkinson, "Fast, automated, N-dimensional phase-unwrapping algorithm," *Magn. Reson. Med.* **49**(1), 193–197 (2003).
3. X. Su and W. Chen, "Fourier transform profilometry: a review," *Opt. Lasers Eng.* **35**(5), 263–284 (2001).
4. C. Zuo et al., "Deep learning in optical metrology: a review," *Light Sci. Appl.* **11**(1), 39 (2022).
5. H. Yu et al., "Phase unwrapping in InSAR: a review," *IEEE Geosci. Remote Sens. Mag.* **7**(1), 40–58 (2019).
6. Y. Lan et al., "Comparative study of DEM reconstruction accuracy between single- and multibaseline InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.* **60**, 1–11 (2022).
7. K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl. Opt.* **21**(14), 2470 (1982).
8. R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," *Radio Sci.* **23**(4), 713–720 (1988).
9. D. J. Bone, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," *Appl. Opt.* **30**(25), 3627 (1991).

10. D. Labrousse, S. Dupont, and M. Berthod, "SAR interferometry: a Markovian approach to phase unwrapping with a discontinuity model," in *Int. Geosci. and Remote Sens. Symp., IGARSS '95. Quant. Remote Sens. for Sci. and Appl.*, IEEE, Vol. 1, pp. 556–558 (1995).
11. D. C. Ghiglia and L. A. Romero, "Minimum LP-norm two-dimensional phase unwrapping," *J. Opt. Soc. Am. A* **13**(10), 1999–2013 (1996).
12. Q. Kemao, "Windowed Fourier transform for fringe pattern analysis," *Appl. Opt.* **43**(13), 2695 (2004).
13. Q. Kemao, "Two-dimensional windowed Fourier transform for fringe pattern analysis: principles, applications and implementations," *Opt. Lasers Eng.* **45**(2), 304–317 (2007).
14. K. Qian, *Windowed Fringe Pattern Analysis*, SPIE Press (2013).
15. M. D. Pritt, "Congruence in least-squares phase unwrapping," in *IGARSS'97. 1997 IEEE Int. Geosci. and Remote Sens. Symp. Proc. Remote Sens. – A Sci. Vision for Sustain. Dev.*, IEEE, Vol. 2, pp. 875–877 (1997).
16. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, Wiley (1998).
17. X. Su and W. Chen, "Reliability-guided phase unwrapping algorithm: a review," *Opt. Lasers Eng.* **42**(3), 245–261 (2004).
18. E. Zappa and G. Busca, "Comparison of eight unwrapping algorithms applied to Fourier-transform profilometry," *Opt. Lasers Eng.* **46**(2), 106–116 (2008).
19. M. Zhao et al., "Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies," *Appl. Opt.* **50**(33), 6214 (2011).
20. K. H. Jin et al., "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.* **26**(9), 4509–4522 (2017).
21. W. Schwartzkopf et al., "Two-dimensional phase unwrapping using neural networks," in *4th IEEE Southwest Symp. Image Anal. and Interpretation*, IEEE Computer Society, pp. 274–277 (2000).
22. G. Dardikman and N. T. Shaked, "Phase unwrapping using residual neural networks," in *Imaging and Appl. Opt. 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, PcAOP)*, OSA, p. CW3B.5 (2018).
23. G. Dardikman, N. A. Turko, and N. T. Shaked, "Deep learning approaches for unwrapping phase images with steep spatial gradients: a simulation," in IEEE Int. Conf. Sci. Electr. Eng. in Israel (ICSEE), IEEE, pp. 1–4 (2018).
24. K. Wang et al., "One-step robust deep learning phase unwrapping," *Opt. Express* **27**(10), 15100 (2019).
25. J. J. He et al., "Deep spatiotemporal phase unwrapping of phase-contrast MRI data," in *Proc Int. Soc. Magn. Reson. Med.*, Vol. 27, p. 1962 (2019).
26. K. Ryu et al., "Development of a deep learning method for phase unwrapping MR images," in *Proc Int. Soc. Magn. Reson. Med.*, Vol. 27, p. 4707 (2019).
27. G. Dardikman-Yoffe et al., "PhUn-Net: ready-to-use neural network for unwrapping quantitative phase images of biological cells," *Biomed. Opt. Express* **11**(2), 1107 (2020).
28. Y. Qin et al., "Direct and accurate phase unwrapping with deep neural network," *Appl. Opt.* **59**(24), 7258 (2020).
29. M. V. Perera and A. De Silva, "A joint convolutional and spatial quad-directional LSTM network for phase unwrapping," in *ICASSP*, pp. 4055–4059 (2021).
30. S. Park, Y. Kim, and I. Moon, "Automated phase unwrapping in digital holography with deep learning," *Biomed. Opt. Express* **12**(11), 7064 (2021).
31. H. Zhou et al., "The PHU-NET: a robust phase unwrapping method for MRI based on deep learning," *Magn. Reson. Med.* **86**(6), 3321–3333 (2021).
32. M. Xu et al., "PU-M-Net for phase unwrapping with speckle reduction and structure protection in ESPI," *Opt. Lasers Eng.* **151**, 106824 (2022).
33. L. Zhou et al., "PU-GAN: a one-step 2-D InSAR phase unwrapping based on conditional generative adversarial network," *IEEE Trans. Geosci. Remote Sens.* **60**, 1–10 (2022).
34. R. Liang et al., "Phase unwrapping using segmentation," Provisional Patent Application patent 62/768,624 (2018).
35. G. E. Spoorthi, S. Gorthi, and R. K. S. S. Gorthi, "PhaseNet: a deep convolutional neural network for two-dimensional phase unwrapping," *IEEE Signal Process. Lett.* **26**(1), 54–58 (2018).
36. J. Zhang et al., "Phase unwrapping in optical metrology via denoised and convolutional segmentation networks," *Opt. Express* **27**(10), 14903 (2019).
37. T. Zhang et al., "Rapid and robust two-dimensional phase unwrapping via deep learning," *Opt. Express* **27**(16), 23173 (2019).
38. C. Wu et al., "Phase unwrapping based on a residual en-decoder network for phase images in Fourier domain Doppler optical coherence tomography," *Biomed. Opt. Express* **11**(4), 1760 (2020).
39. G. E. Spoorthi, R. K. S. S. Gorthi, and S. Gorthi, "PhaseNet 2.0: phase unwrapping of noisy data based on deep learning approach," *IEEE Trans. Image Process.* **29**, 4862–4872 (2020).
40. Z. Zhao et al., "Phase unwrapping method for point diffraction interferometer based on residual auto encoder neural network," *Opt. Lasers Eng.* **138**, 106405 (2020).
41. S. Zhu et al., "Phase unwrapping in ICF target interferometric measurement via deep learning," *Appl. Opt.* **60**(1), 10 (2021).
42. K. S. Vengala, N. Paluru, and R. K. S. S. Gorthi, "3D deformation measurement in digital holographic interferometry using a multitask deep learning architecture," *J. Opt. Soc. Am. A* **39**(1), 167 (2022).
43. K. S. Vengala, V. Ravi, and G. R. K. S. Subrahmanyam, "A multi-task learning for 2D phase unwrapping in fringe projection," *IEEE Signal Process. Lett.* **29**, 797–801 (2022).
44. J. Zhang and Q. Li, "EESANet: edge-enhanced self-attention network for two-dimensional phase unwrapping," *Opt. Express* **30**(7), 10470 (2022).
45. K. Yan et al., "Wrapped phase denoising using convolutional neural networks," *Opt. Lasers Eng.* **128**, 105999 (2020).
46. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *California Univ. San Diego La Jolla Inst for Cognitive Science*, Elsevier, pp. 399–421 (1988).
47. K. He et al., "Deep residual learning for image recognition," in IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR), IEEE, pp. 770–778 (2016).
48. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab et al., Eds., pp. 234–241, Springer International Publishing (2015).
49. L.-C. Chen et al., "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, pp. 801–818 (2018).
50. T. Pohlen et al., "Full-resolution residual networks for semantic segmentation in street scenes," in IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR), IEEE, pp. 3309–3318 (2017).
51. K. Wang et al., "Y-Net: a one-to-two deep learning framework for digital holographic reconstruction," *Opt. Lett.* **44**(19), 4765 (2019).
52. T. Nguyen et al., "Deep learning approach for Fourier ptychography microscopy," *Opt. Express* **26**(20), 26470 (2018).
53. Y. Wu et al., "Extended depth-of-field in holographic imaging using deep-learning-based autofocusing and phase recovery," *Optica* **5**(6), 704 (2018).
54. N. Borhani et al., "Learning to see through multimode fibers," *Optica* **5**(8), 960 (2018).
55. S. K. Devalla et al., "DRUNET: a dilated-residual U-Net deep learning network to segment optic nerve head tissues in optical coherence tomography images," *Biomed. Opt. Express* **9**(7), 3244 (2018).
56. G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica* **6**(8), 921 (2019).

57. Y. Rivenson et al., "PhaseStain: the digital staining of label-free quantitative phase microscopy images using deep learning," *Light Sci. Appl.* **8**(1), 23 (2019).
58. Y. Rivenson et al., "Virtual histological staining of unlabelled tissue-autofluorescence images via deep learning," *Nat. Biomed. Eng.* **3**(6), 466–477 (2019).
59. Y. Rivenson, Y. Wu, and A. Ozcan, "Deep learning in holography and coherent imaging," *Light Sci. Appl.* **8**(1), 85 (2019).
60. S. Feng et al., "Fringe pattern analysis using deep learning," *Adv. Photon.* **1**(2), 025001 (2019).
61. H. Wang et al., "Deep learning enables cross-modality super-resolution in fluorescence microscopy," *Nat. Methods* **16**(1), 103–110 (2019).
62. J. Zhao et al., "Deep-learning cell imaging through Anderson localizing optical fiber," *Adv. Photon.* **1**(6), 066001 (2019).
63. W. Yin et al., "Temporal phase unwrapping using deep learning," *Sci. Rep.* **9**, 20175 (2019).
64. Z. Ren, Z. Xu, and E. Y. Lam, "End-to-end deep learning framework for digital holographic reconstruction," *Adv. Photon.* **1**(1), 016004 (2019).
65. H. Yu et al., "Dynamic 3-D measurement based on fringe-to-fringe transformation using deep learning," *Opt. Express* **28**(7), 9405–9418 (2020).
66. J. Tang et al., "RestoreNet: a deep learning framework for image restoration in optical synthetic aperture imaging system," *Opt. Lasers Eng.* **139**, 106463 (2020).
67. K. Wang et al., "Transport of intensity equation from a single intensity image via deep learning," *Opt. Lasers Eng.* **134**, 106233 (2020).
68. K. Wang et al., "Y4-Net: a deep learning solution to one-shot dual-wavelength digital holographic reconstruction," *Opt. Lett.* **45**(15), 4220 (2020).
69. J. Qian et al., "Single-shot absolute 3D shape measurement with deep-learning-based color fringe projection profilometry," *Opt. Lett.* **45**(7), 1842–1845 (2020).
70. M. Lyu et al., "Learning-based lensless imaging through optically thick scattering media," *Adv. Photon.* **1**(3), 036002 (2019).
71. J. Lim, A. B. Ayoub, and D. Psaltis, "Three-dimensional tomography of red blood cells using deep learning," *Adv. Photon.* **2**(2), 026001 (2020).
72. K. Wang et al., "Deep learning wavefront sensing and aberration correction in atmospheric turbulence," *PhotoniX* **2**(1), 8 (2021).
73. J. Wu, L. Cao, and G. Barbastathis, "DNN-FZA camera: a deep learning approach toward broadband FZA lensless imaging," *Opt. Lett.* **46**(1), 130 (2021).
74. J. Wu et al., "High-speed computer-generated holography using an autoencoder-based deep neural network," *Opt. Lett.* **46**(12), 2908 (2021).
75. S. Zheng et al., "Incoherent imaging through highly nonstatic and optically thick turbid media based on neural network," *Photon. Res.* **9**(5), B220 (2021).
76. Y. Wu et al., "Dense-U-net: dense encoder–decoder network for holographic imaging of 3D particle fields," *Opt. Commun.* **493**, 126970 (2021).
77. M. Liao et al., "Deep-learning-based ciphertext-only attack on optical double random phase encryption," *Opto-Electron. Adv.* **4**(5), 200016 (2021).
78. C. Szegedy et al., "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, IEEE, pp. 2818–2826 (2016).
79. J. Di et al., "Dual-wavelength common-path digital holographic microscopy for quantitative phase imaging based on lateral shearing interferometry," *Appl. Opt.* **55**(26), 7287 (2016).
80. Y. Li et al., "Quantitative phase microscopy for cellular dynamics based on transport of intensity equation," *Opt. Express* **26**(1), 586 (2018).
81. C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.* **27**(3), 379–423 (1948).
82. M. Deng et al., "On the interplay between physical and content priors in deep learning for computational imaging," *Opt. Express* **28**(16), 24152 (2020).
83. K. Qian, "Updated WFT for fringe analysis," https://www.mathworks.com/matlabcentral/fileexchange/64962-updated-wft-for-fringe-analysis (2017).
84. I. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst. 27* (2014).
85. S. Du et al., "Affine iterative closest point algorithm for point set registration," *Pattern Recognit. Lett.* **31**(9), 791–799 (2010).
86. M. Xin et al., "A robust cloud registration method based on redundant data reduction using backpropagation neural network and shift window," *Rev. Sci. Instrum.* **89**(2), 024704 (2018).
87. A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Advances in Neural Information Processing Systems*, I. Guyon et al., Eds., Curran Associates, Inc. (2017).
88. Y. Chen et al., "Dynamic convolution: attention over convolution kernels," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 11030–11039 (2020).
89. O. Oktay et al., "Attention U-net: learning where to look for the pancreas," arXiv180403999 Cs (2018).
90. J. Yosinski et al., "How transferable are features in deep neural networks?," in *Adv. in Neural Inform. Process. Syst.*, Z. Ghahramani et al., Eds., Curran Associates, Inc. (2014).
91. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 9446–9454 (2018).
92. F. Yang et al., "Robust phase unwrapping via deep image prior for quantitative phase imaging," *IEEE Trans. Image Process.* **30**, 7025–7037 (2021).
93. F. Wang et al., "Phase imaging with an untrained neural network," *Light Sci. Appl.* **9**(1), 77 (2020).
94. D. Yang et al., "Dynamic coherent diffractive imaging with a physics-driven untrained learning method," *Opt. Express* **29**(20), 31426 (2021).
95. F. Niknam, H. Qazvini, and H. Latifi, "Holographic optical field recovery using a regularized untrained deep decoder network," *Sci. Rep.* **11**, 10903 (2021).

Kaiqiang Wang received the bachelor's and PhD degrees from the Northwestern Polytechnical University in 2016 and 2022, respectively. His research interests include computational imaging and deep learning.

Qian Kemao is currently an associate professor at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his bachelor's, master's, and PhD degrees from the University of Science and Technology of China in 1994, 1997, and 2000, respectively. He has published more than 150 scientific articles and a monograph. His research interests include optical metrology, image processing, computer vision, and computer animation.

Jianglei Di is currently a professor at the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. He received his BS, MS, and PhD degrees from NPU in 2004, 2007, and 2012, respectively. His research interests include digital holography, optical information processing, optical precision measurement and deep learning.

Jianlin Zhao is currently a professor at the School of Physical Science and Technology, Northwestern Polytechnical University (NPU), Xi'an, China. He received his BS and MS degrees from NPU in 1981 and 1987, respectively. He received his PhD in optics from Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, in 1998. He is the director of Shaanxi Key Laboratory of Optical Information Technology, and MOE Key Laboratory of Material Physics and Chemistry under Extraordinary Conditions. His research interests include light field manipulation, imaging, information processing, and applications.