Research Article

ADVANCED PHOTONICS NEXUS

Deep learning phase recovery: data-driven, physics-driven, or a combination of both?

Kaiqiang Wang* and **Edmund Y. Lam***
University of Hong Kong, Department of Electrical and Electronic Engineering, Hong Kong, China

Abstract. Phase recovery, calculating the phase of a light wave from its intensity measurements, is essential for various applications, such as coherent diffraction imaging, adaptive optics, and biomedical imaging. It enables the reconstruction of an object's refractive index distribution or topography as well as the correction of imaging system aberrations. In recent years, deep learning has been proven to be highly effective in addressing phase recovery problems. The two most direct deep learning phase recovery strategies are data-driven (DD) with supervised learning mode and physics-driven (PD) with self-supervised learning mode. DD and PD achieve the same goal in different ways yet there is a lack of necessary research to reveal similarities and differences. Therefore, we comprehensively compare these two deep learning phase recovery strategies in terms of time consumption, accuracy, generalization ability, ill-posedness adaptability, and prior capacity. What is more, we propose a co-driven strategy of combining datasets and physics for the balance of high- and low-frequency information.

Keywords: phase recovery; deep learning; computational imaging.

Received May 9, 2024; revised manuscript received Jul. 22, 2024; accepted for publication Aug. 12, 2024; published online Sep. 13, 2024.

[DOI: 10.1117/1.APN.3.5.056006]

1 Introduction

Phase recovery refers to a class of methods that recover the phase of light waves from intensity measurements.[1] It is active in various fields of imaging and detection, such as in bioimaging for obtaining the refractive index or thickness distribution of tissues or cells,[2] in adaptive optics for characterizing aberrant wavefronts,[3] in coherent diffraction imaging for detecting the structural information of nanomolecules,[4] and in material inspection for measuring surface profiles.[5]

Since optical detectors, such as charge-coupled device sensors, can only record the intensity/amplitude but lose the phase, one has to recover the phase from the recorded intensity indirectly. And precisely because of the loss of the phase, it is ill-posed to directly calculate the phase on the object plane from the only amplitude on the measurement plane through the forward physical model. On the one hand, the phase can be iteratively retrieved from intensity measurements with prior knowledge, i.e., phase retrieval.[6] On the other hand, by incorporating additional information, this problem can be transformed into a well-posed one and solved directly, such as holography or interferometry with reference light,[7,8] Shack-Hartmann wavefront sensing with micro-lens arrays,[9,10] and the transport of intensity equation with multiple through-focus intensity images.[11,12]

In recent years, deep learning, with artificial neural networks as the carrier, has brought new solutions to phase recovery. One of the most direct ways is to train neural networks to learn the mapping relationship from intensity measurements to the light wave phase.[1,13,14] On the one hand, the training of neural networks can be driven by paired input-label datasets as implicit prior knowledge ("implicit prior"); these are called data-driven (DD) strategies (see the upper part of Fig. 1).[1] On the other hand, forward physical models can be used as explicit prior knowledge ("explicit prior") to drive the training of neural networks with input-only datasets; these are called physics-driven (PD) strategies (see the lower part of Fig. 1).[1] In addition, neural networks can also indirectly participate in the process of phase recovery including pre-processing, in-processing (physics-connect-network, network-in-physics, and physics-in-network), and post-processing.[1] Compared with classic phase recovery methods that mainly rely on physical models, deep learning methods additionally introduce prior knowledge from datasets and neural network structures to improve efficiency.

*Address all correspondence to Kaiqiang Wang, kqwang.optics@gmail.com; Edmund Y. Lam, elam@eee.hku.hk

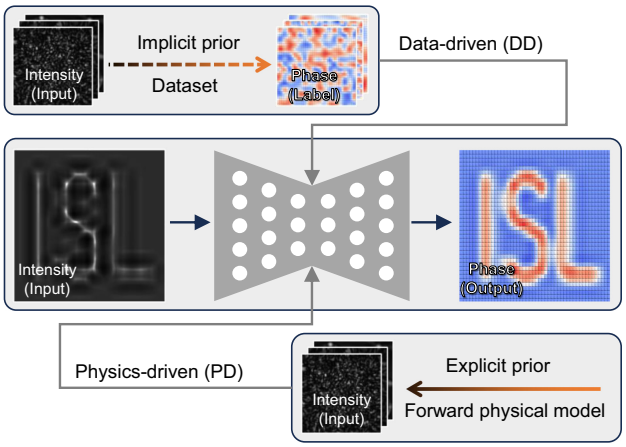

Fig. 1 Phase recovery network training with DD and PD strategies.

Sinha et al.[15] first demonstrated DD phase recovery with paired diffraction-phase datasets, obtained by recording diffraction images of virtual phase objects loaded on a spatial light modulator. Subsequently, DD phase recovery was successively extended to in-line holography,[16] coherent diffraction imaging,[17] Fourier ptychography,[18] off-axis holography,[19] Shack-Hartmann wavefront sensing,[20] transport of intensity equation,[21] optical diffraction tomography,[22] and electron diffractive imaging.[23] In addition, several studies focused on more efficient neural network structures for phase recovery, such as the Bayesian neural network,[24] generative adversarial network,[25] Y-Net,[26,27] residual capsule network,[28] recurrent neural network,[29] Fourier imager network,[30,31] and neural architecture search.[32] Some studies also used DD methods for pre- or post-processing of phase recovery, such as defocus distance prediction,[33] resolution enhancement,[34] phase unwrapping,[35] and classification.[36,37]

The idea of PD phase recovery was first introduced by Boominathan et al.[38] in their simulation work on Fourier ptychography. Wang et al.[39] first experimentally used PD to iteratively infer the phase of a phase-only object from its diffraction image directly on an untrained/initialized neural network. Afterward, it was subsequently extended to the cases of unknown defocus distances,[40] dual wavelengths,[41] and complex-valued amplitude objects.[42,43] In the quest for faster inference times, PD and a large number of intensity measurements were used for neural network pre-training.[42–46] Further, refinement of pre-trained neural networks by PD achieved higher accuracy with lower inference time.[47,48] It should be noted that the PD strategies mentioned here do not include methods that use random vectors or matrices as the inputs of neural networks. For the specific differences, please refer to the italicized part on page 22 of Ref. 1.

DD and PD achieve the same goal in different ways and are being studied in different contexts to achieve efficient phase recovery. Therefore, it is necessary and meaningful to compare them under the same context. In this paper, we introduce the principles of DD and PD and comparatively study them in terms of time consumption, accuracy, generalization ability, ill-posedness adaptability, and prior capacity. We also combine DD and PD as a co-driven (CD) strategy to train neural networks for high- and low-frequency information balance. What is more, to facilitate readers to get started with the deep learning phase recovery quickly, we release the demonstrations of DD, PD, and CD which are available in a Github repository at: https://github.com/kqwang/DLPR

2 Principles and Methods

Here, we consider a classic phase recovery paradigm, recovering the phase or complex-valued amplitude of a light wave from its in-line hologram (diffraction pattern). For an object illuminated by a coherent plane wave, its hologram can be written as

$$H = G(A, P), \tag{1}$$

where H is the hologram, A is the amplitude of the light wave, P is the phase of the light wave, and $G(\cdot)$ is the forward propagation function. For a phase object, we assume $A = 1$. Then, the purpose of phase recovery is to formulate the inverse mapping of $G(\cdot)$:

$$P = G^{-1}(H). \tag{2}$$

With a supervised learning mode, DD trains neural networks with paired hologram-phase datasets $S_{H-P} = \{(H_i, P_i), i = 1, \ldots, N\}$ as an implicit prior to learn this inverse mapping:[15]

$$f_{\omega^*} = \arg\min_{f_\omega} \sum_{i=1}^{N} \|f_\omega(H_i) - P_i\|_2^2, \quad \forall\, (H_i, P_i) \in S_{H-P}, \tag{3}$$

where $\|\cdot\|_2^2$ denotes the square of the l_2-norm (or other distance functions) and f_ω is a neural network with trainable parameters ω, such as weights and biases. When the optimization is complete, the trained neural network f_{ω^*} is used as an inverse mapper to infer the corresponding phase $\hat{P}_x$ from its hologram H_x of an unseen object that is not in the training dataset:

$$\hat{P}_x = f_{\omega^*}(H_x). \tag{4}$$

A visual representation of DD is shown in Fig. 2, in which holograms and phases are used as the input and ground truth (GT) of the neural network, respectively. The training dataset, collected through experiments or numerical simulations, typically contains paired data from thousands to hundreds of thousands. The training stage usually lasts for hours or even days but only takes one time. After that, the trained neural network quickly infers the phase of the unseen object after being fed its hologram.

For physical processes that can be well modeled, such as phase recovery, PD is another available strategy. With a self-supervised learning mode, PD uses a numerical propagation $G(\cdot)$ as an explicit prior to drive the training or inference of neural networks (Fig. 3). Different from DD, which calculates the loss function in the phase domain, PD converts the network output from the phase domain to the hologram domain via numerical propagation $G(\cdot)$ and then calculates the loss function. This numerical propagation $G(\cdot)$ can be utilized to optimize the neural network in three ways: untrained PD (uPD),[39] trained PD (tPD),[45] and tPD with refinement (tPDr).[47]

With the driving of the numerical propagation $G(\cdot)$, uPD iteratively optimizes an initialized neural network $f_\omega(\cdot)$ to directly infer the phase $\hat{P}_x$ of an unseen object from its hologram H_x [Fig. 3(a)]:

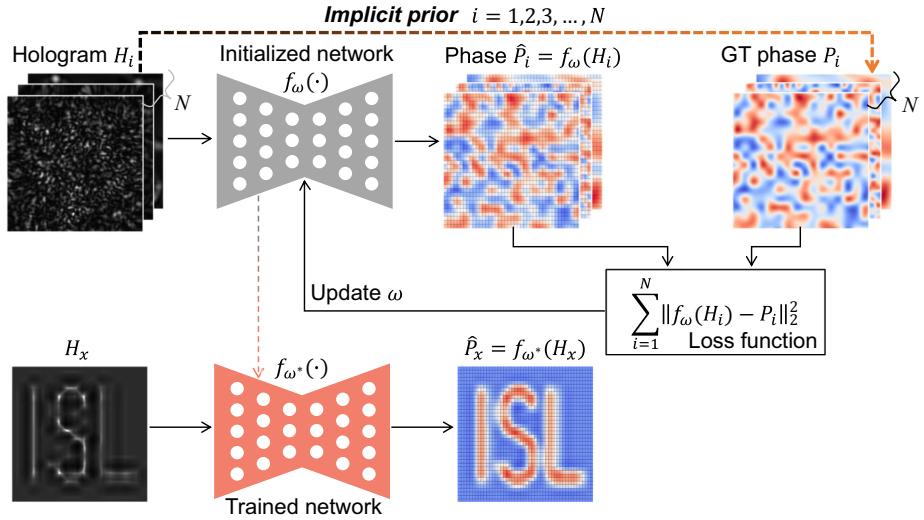

Fig. 2 Description of DD deep learning phase recovery methods.

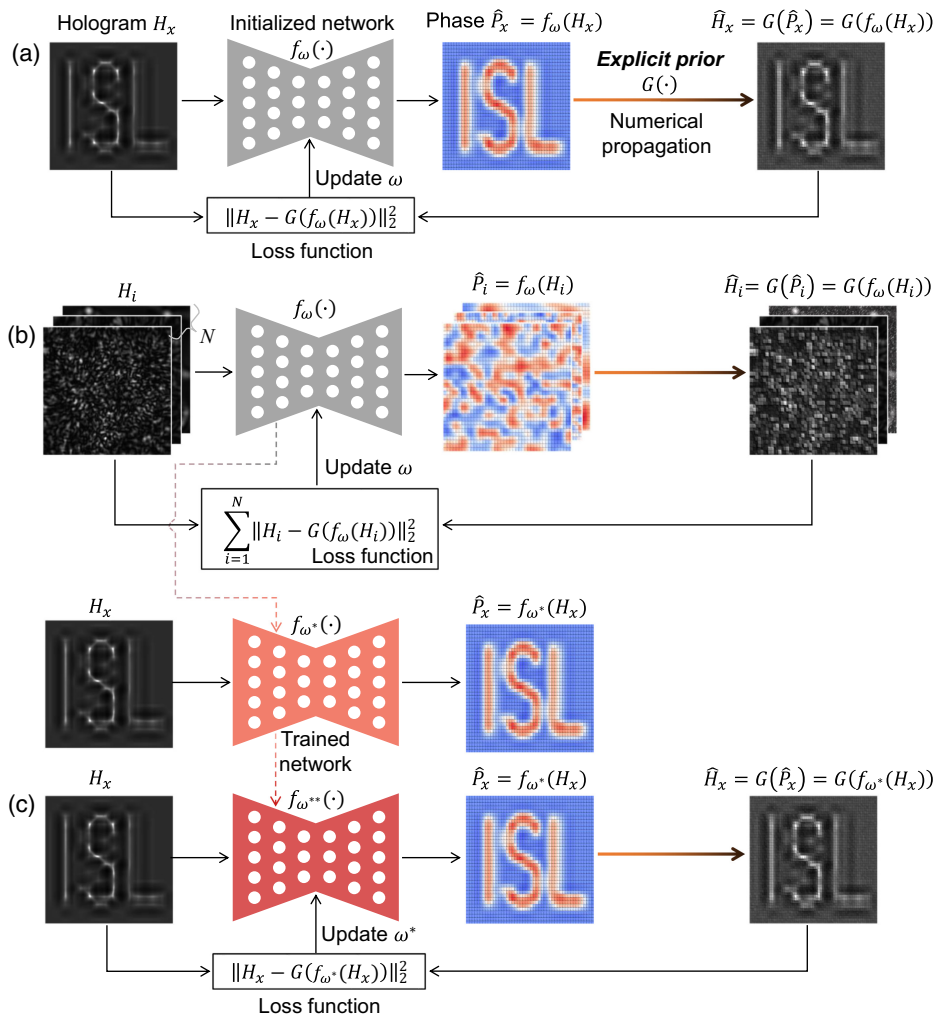

Fig. 3 Description of PD deep learning phase recovery methods. (a) Network inference for the uPD. (b) Network training and inference for the tPD. (c) Network training and inference for the tPDr.

Table 1 Summary of DD, uPD, tPD, and tPDr.

Strategy	Physics requirement	Dataset requirement	Inference cycles	Learning mode
DD	No	Hologram-phase dataset	One time	Supervised
uPD	Numerical propagation	No	Multi times	Self-supervised
tPD	Numerical propagation	Hologram-only dataset	One time	Self-supervised
tPDr	Numerical propagation	Hologram-only dataset	Multi times	Self-supervised

$$\begin{cases} f_{\omega^*} = \arg\min_{f_\omega} \|G[f_\omega(H_x)] - H_x\|_2^2 \\ \hat{P}_x = f_{\omega^*}(H_x). \end{cases} \tag{5}$$

The most significant advantage of uPD is that it does not require any dataset to pre-process the neural network before inferences.

In tPD, the numerical propagation $G(\cdot)$ is employed to train the neural network $f_\omega(\cdot)$ with intensity-only training dataset $S_H = \{(H_i), i = 1, \ldots, N\}$ as input, and then the trained neural network f_{ω^*} infers the phase $\hat{P}_x$ of an unseen object from its hologram H_x [Fig. 3(b)]:

$$\begin{cases} f_{\omega^*} = \arg\min_{f_\omega} \sum_{i=1}^{N} \|G[f_\omega(H_i)] - H_i\|_2^2, \quad \forall\ (H_i) \in S_H \\ \hat{P}_x = f_{\omega^*}(H_x). \end{cases} \tag{6}$$

Comparing Eqs. (3) and (6), we can find that the working modes of tPD and DD are similar. However, due to the use of numerical propagation $G(\cdot)$, the training dataset for tPD only requires a large number of holograms without the corresponding phase as GT.

As a strategy combining uPD and tPD, tPDr iteratively fine-tunes the tPD trained neural network $f_{\omega^*}(\cdot)$ on the hologram of the unseen object [Fig. 3(c)]:

$$\begin{cases} f_{\omega^{**}} = \arg\min_{f_\omega^*} \|G[f_{\omega^*}(H_x)] - H_x\|_2^2 \\ \hat{P}_x = f_{\omega^{**}}(H_x). \end{cases} \tag{7}$$

In addition, some methods use both forward physical models and DD neural networks for phase recovery. On the one hand, some methods first use forward physical models to recover preliminary phases from holograms and then use DD neural networks to either remove unwanted components[49,50] or perform resolution enhancement[51,52] or convert imaging modalities.[53] On the other hand, some methods use DD neural networks to generate holograms with different propagation distances from a hologram and then recover the phase using iterative algorithms based on forward physical models.[54] There is also an interesting way to introduce DD into PD in the form of a generative adversarial network for phase recovery.[55]

For the sake of clarity, we summarize DD, uPD, tPD, and tPDr according to their requirements for the physical model, the training dataset, the number of cycles needed for inference, and the learning mode in Table 1.

3 Results and Discussion

To avoid unnecessary distraction factors, all datasets used for comparison are generated through numerical simulation based on ImageNet, LFW, and MNIST; see Appendix A. ImageNet represents highly complex dense samples, LFW represents moderately complex dense samples, and MNIST represents simple sparse samples. Given its ubiquity in computational imaging, all methods use the same U-Net-based neural network, the specific structure of which is described in the Supplementary Material of Ref. 56. The implementation of the neural network is set uniformly; see Appendix B. The average peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are used to quantify the inference accuracy.

3.1 Comparison of Time Consumption and Accuracy

In this section, ImageNet is used for dataset generation. We summarize the training settings and inference evaluation of DD, uPD, tPD, and tPDr in Table 2.

In terms of time consumption, DD, tPD, and tPDr all require pre-training before inference, thus consuming hours or even more for neural network optimization, whereas uPD performs inference for the tested sample directly on an initialized neural network. During the inference stage of DD and tPD, the hologram of the tested sample passes through the trained neural network once in one second, while the inference process for uPD and tPDr takes several minutes for iteration.

As for the inference accuracy, the PSNR and SSIM of DD and tPD that do quick inference once after pre-training are basically the same, and both are significantly lower than uPD and tPDr, which do inference multiple times. Due to the prior knowledge introduced in the pre-training stage, the initial inference of tPDr is closer to the target solution, which makes it obtain the same accuracy with shorter inference cycles than uPD. Specifically, with comparable accuracy, the inference time of tPDr is 1/10th that of uPD.

Table 2 Training settings and inference evaluation of DD, uPD, tPD, and tPDr.

Strategy	Training datasets	Inference cycles	Inference time (s)	PSNR ↑	SSIM ↑
DD	10,000 pairs	1	0.02	19.9	0.68
uPD	0	10,000	800	25.6	0.94
tPD	10,000 inputs	1	0.02	18.5	0.69
tPDr	10,000 inputs	1000	80	25.1	0.93

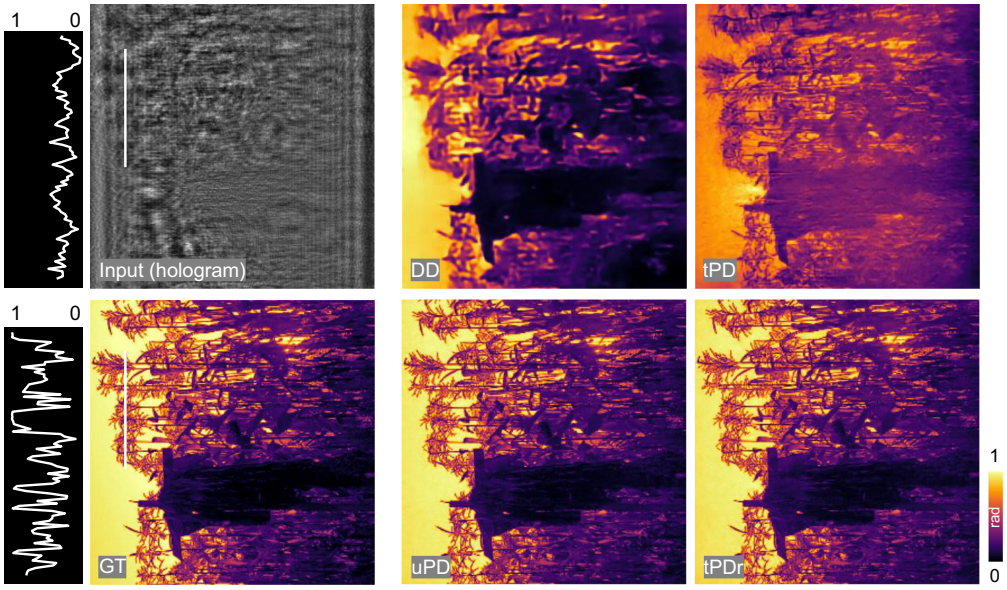

Fig. 4 Inference results of DD, uPD, tPD, and tPDr.

Although it has the same accuracy index (Table 2), the inference result of tPD shows better high-frequency detailed information while that of DD shows better low-frequency background information (Fig. 4). According to the frequency principle, deep neural networks are more inclined to learn low-frequency information in the data.[57] DD learns the hologram-phase mapping relationship through the loss function in the phase domain, while PD uses numerical propagation to transfer it from the phase domain to the hologram domain. On the one hand, as shown in the white curve on the left side of Fig. 4, the high-frequency phase information (steeper curve) is recorded in the diffraction fringes of the hologram, which contains a more balanced high- and low-frequency information (smoother curve). This is more favorable for PD to learn high-frequency phase information from the loss function in the hologram domain. On the other hand, the low-frequency phase causes only little contrast in the hologram, making it difficult for PD to learn low-frequency phase information, especially the plane background phase.

In order to balance the high- and low-frequency phase information learned by the neural network, we propose to use both the dataset and physics for the neural network training, named CD. The loss function of CD is derived from the weighted sum of the DD term and PD term:

$$\begin{cases} f_{\omega^*} = \arg\min\limits_{f_\omega} \sum_{i=1}^{N} \alpha \|f_\omega(H_i) - P_i\|_2^2 + \|G[f_\omega(H_i)] - H_i\|_2^2, \\ \forall\ (H_i, P_i) \in S_{H-P}, \end{cases} \tag{8}$$

where α is the weight used to control the contribution of the DD term and PD term, which is set to 0.3. As shown in Fig. 5, compared to the low-frequency-tendency DD and high-frequency-tendency tPD, CD takes into account both the high-frequency phase (see the blue box) and low-frequency phase (see the green box). It should be noted that we only compared CD with DD and tPD since they all go through the neural network once for inference.

Interestingly, by comparing the inference results of holograms under different propagation distances (see Fig. S1 in the Supplementary Material), we find that DD has a higher tolerance for defocus distance than tPD. This is most likely due to the fact that the loss function used by tPD for the neural network training is calculated in the hologram domain, and thus it is more sensitive to changes in defocus holograms than DD. In addition, the CD's sensitivity to defocus distance is neutralized.

3.2 Comparison of Generalization Ability

To compare the generalization ability of DD and tPD, ImageNet, LFW, and MNIST are used to generate datasets for neural network training and cross-inference, respectively. ImageNet represents dense samples, MNIST represents sparse samples, and LFW is somewhere in between. In Fig. 6, we show the cross-inference results and their absolute error maps of a sample from ImageNet, LFW, and MNIST, and attach the average SSIM on the testing dataset below each result.

Overall, the dataset is the main factor affecting the generalization ability of the trained neural network. Specifically, the neural networks trained by ImageNet and LFW generally perform better on all three testing datasets, while the neural networks trained by MNIST can only infer the overall distribution of ImageNet and LFW but lack detailed information. Admittedly, MNIST itself lacks detailed information, so it is reasonable that neural networks trained with it would not be able to fully infer detailed information about ImageNet and LFW. In this extreme case, tPD is significantly better than DD, both in terms of inference results and SSIM. As can be shown in Fig. 6, tPD infers more detailed information than DD (marked by the green arrow). Nonetheless, these results are sufficient to prove the strong generalization ability of DD and tPD, because MNIST used for training is very sparse handwritten digits with monotonous features, but the trained neural network can still do inference for the complex and feature-rich samples in ImageNet

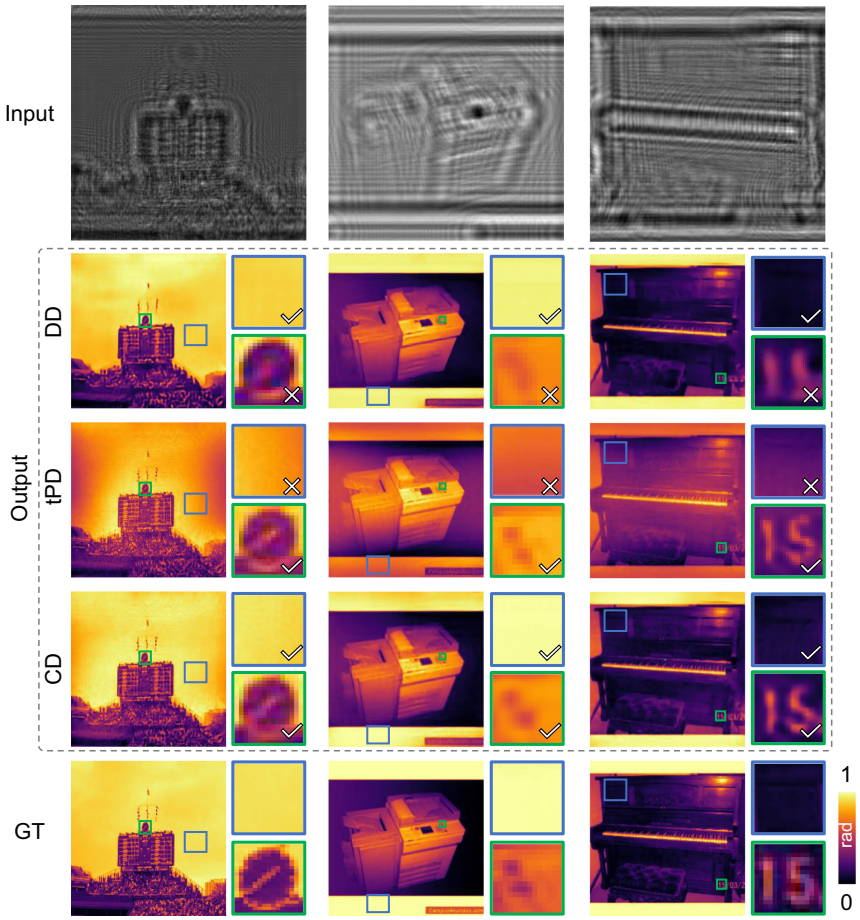

Fig. 5 Results of DD, tPD, and CD. The blue box represents low-frequency information and the green box represents high-frequency information.

and LFW. Another thing worth noting is that for the case of using neural networks trained by ImageNet and LFW to infer MNIST, although the inference results of both tPD and DD appear to be ideal, the SSIM of tPD is much lower than that of DD. As can be seen from the absolute error maps (marked by the yellow arrow), the error in the background part of tPD is relatively larger than that of DD, which confirms a conclusion in Sec. 3.1 that tPD is not good at low-frequency phase information, especially the plane background phase.

3.3 Comparison of Ill-Posedness Adaptability

Let us consider a more ill-posed case of using a neural network to simultaneously infer phase and amplitude from a hologram. In dataset generation, ImageNet, LFW, and MNIST are used to get samples containing phase and amplitude, and the corresponding holograms are calculated through numerical propagation. Given that the neural network needs to output both phase and amplitude, we modified the original U-Net by paralleling another up-sampling path to build a Y-Net.[26] The way tPD trains the neural network has not changed, except that there is an amplitude term in the loss function:

$$\begin{cases} f_{\omega^*}^{P,A} = \arg\min\limits_{f_\omega^{P,A}} \|G[f_\omega^{P,A}(H_x)] - H_x\|_2^2 \\ \hat{P}_x, \hat{A}_x = f_{\omega^*}^{P,A}(H_x), \end{cases} \tag{9}$$

where $f_\omega^{P,A}(\cdot)$ denotes the Y-Net that outputs phase and amplitude simultaneously. The loss function of DD is derived by weighted summation of the phase term and amplitude term:

$$\begin{cases} f_{\omega^*}^{P,A} = \arg\min\limits_{f_\omega^{P,A}} \sum_{i=1}^{N} \|f_\omega^P(H_i) - P_i\|_2^2 + \beta\|f_\omega^A(H_i) - A_i\|_2^2 \\ \hat{P}_x, \hat{A}_x = f_{\omega^*}^{P,A}(H_x), \end{cases} \tag{10}$$

where $f_\omega^P(\cdot)$ and $f_\omega^A(\cdot)$ denote the phase path and amplitude path of Y-Net, respectively, and β is the weight used to control the contribution of the phase term and amplitude term, which is set to 0.1.

The inference results of DD and tPD with single hologram input are shown in the blue part of Fig. 7. DD can infer the phase and amplitude at the same time because the implicit mapping relationship from the holograms to phase and amplitude is completely included in the paired dataset used for the network training. As for tPD, obvious artifacts appear in the inference results, and its SSIM is reduced accordingly. This means that, although there are many undesirable components in the inference result, the hologram corresponding to this non-ideal phase and the amplitude matches the hologram of the sample. That is, the situation of using a hologram to infer both phase and amplitude simultaneously is severely ill-posed for tPD.

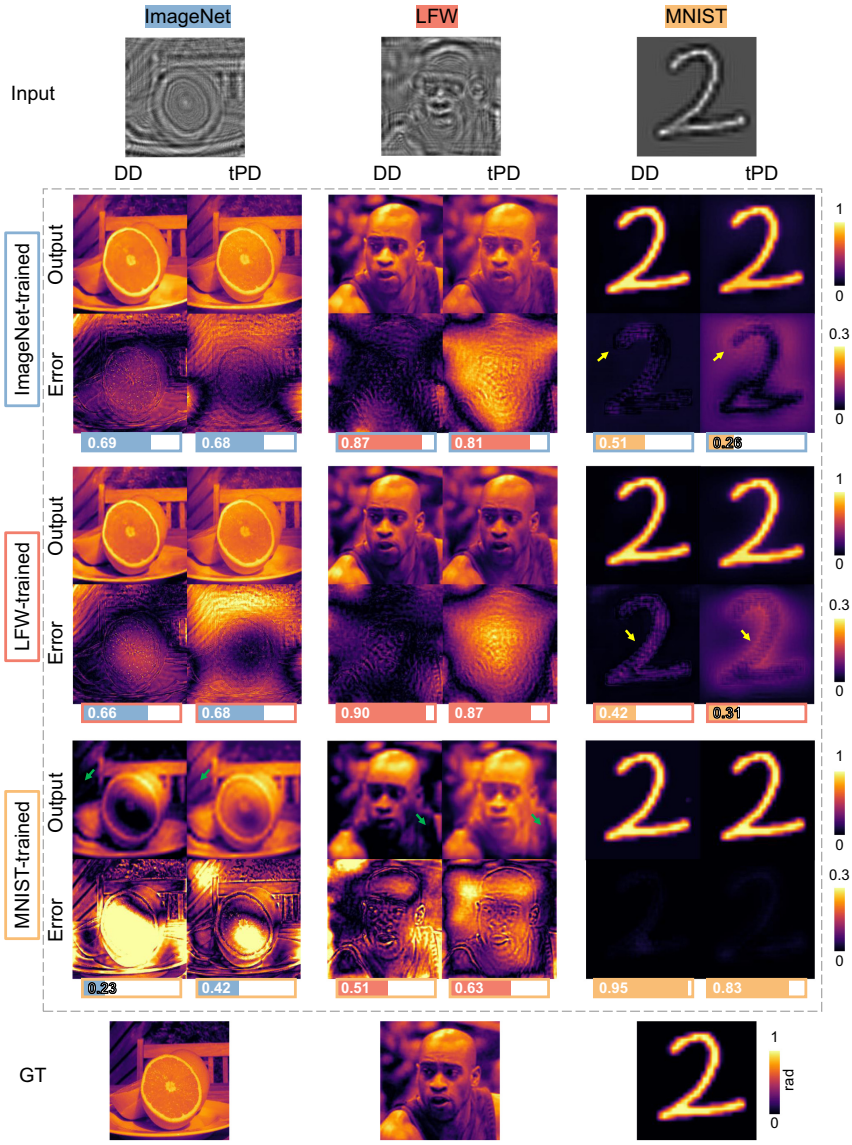

Fig. 6 Cross-inference results of DD and tPD for the datasets of ImageNet, LFW, and MNIST. The metric below each result is the average SSIM for that testing dataset.

Here, we show two solutions for this ill-posedness of tPD. For one thing, we introduce an aperture constraint in the sample plane to reduce the difficulty of tPD phase recovery:[42]

$$\begin{cases} f_{\omega^*}^{P,A} = \arg\min_{f_\omega^{P,A}} \|G[f_\omega^{P,A}(H_x)] - H_x\|_2^2 \\ \qquad + \|f_\omega^A(H_x)\cdot[1-C(r)] - 0_{N\times N}\|_2^2 \\ \hat{P}_x, \hat{A}_x = f_{\omega^*}^{P,A}(H_x), \end{cases} \tag{11}$$

where $C(r)$ is the aperture constraint with the radius r, which is set to 80 pixel, and $0_{N\times N}$ denotes the zero matrix of size $N \times N$, where N is set to 256. After introducing aperture constraints, the inference results of tPD for the three datasets are improved to varying degrees (see the red part of Fig. 7). MNIST has the largest improvement, followed by LFW, and ImageNet has such limited improvement. This means that the aperture constraint works well for simple cases with less information but can hardly deal with more difficult samples. For another thing to further reduce the ill-posedness of tPD, we introduce more prior knowledge by using multiple holograms with different defocus distances as network inputs.[45] In this case, the loss function contains three terms corresponding to different defocus distances:

$$\begin{cases} f_{\omega^*}^{P,A} = \arg\min_{f_\omega^{P,A}} \|G^{z_1}[f_\omega^{P,A}(H_x^{z_1}, H_x^{z_2}, H_x^{z_3})] - H_x^{z_1}\|_2^2 \\ \qquad + \|G^{z_2}[f_\omega^{P,A}(H_x^{z_1}, H_x^{z_2}, H_x^{z_3})] - H_x^{z_2}\|_2^2 \\ \qquad + \|G^{z_3}[f_\omega^{P,A}(H_x^{z_1}, H_x^{z_2}, H_x^{z_3})] - H_x^{z_3}\|_2^2 \\ \hat{P}_x, \hat{A}_x = f_{\omega^*}^{P,A}(H_x^{z_1}, H_x^{z_2}, H_x^{z_3}), \end{cases} \tag{12}$$

where $G^{z_1}(\cdot)$, $G^{z_2}(\cdot)$, and $G^{z_3}(\cdot)$ donate the numerical propagation of different distances, and $H_x^{z_1}, H_x^{z_2}$, and $H_x^{z_3}$ donate the holograms with different defocus distances, where z_1, z_2, z_3

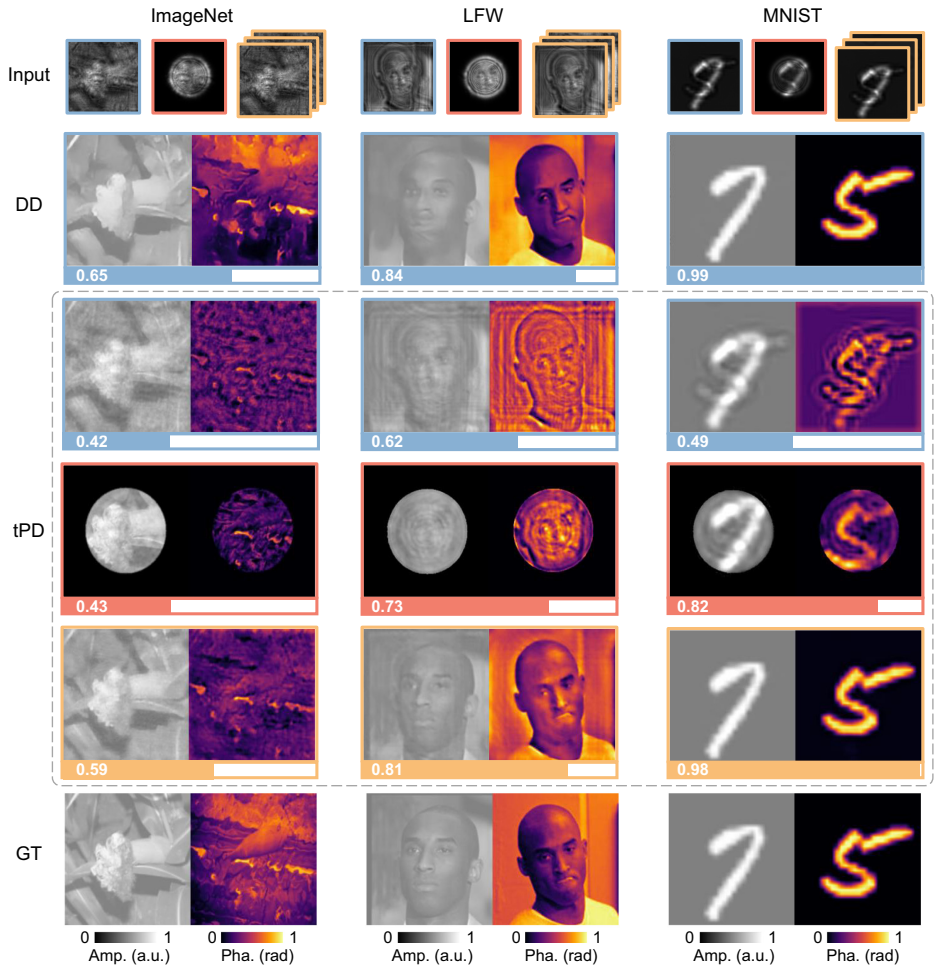

Fig. 7 Ill-posedness adaptability test of DD and tPD. Blue part represents a single hologram as the network input, red part represents a single hologram with aperture constraints as the network input, and yellow part represents multiple holograms as the network input.

are set to 20, 40, and 60 mm, respectively. Compared to a single hologram input, two more holograms introduce sufficient prior knowledge for tPD, resulting in a significant improvement in the trained neural network, both for the simple MNIST and the complex LFW and ImageNet (see the yellow part of Fig. 7).

3.4 Comparison of Prior Capacity

tPD uses numerical propagation as an explicit prior to train the neural network, so the neural network learns priors from numerical propagation. DD trains a neural network with paired datasets, which means that the neural network learns all implicit priors contained in the dataset even if it is outside the numerical propagation. For example, in the presence of imaging aberration, there will be both sample and aberration information in the hologram. Here, we use ImageNet as the sample phase and a random phase generated by the random matrix enlargement[35,56] as the aberration phase to generate a dataset for the comparison of DD and tPD. The process of dataset generation and network training is shown in Fig. 8, where blue represents the dataset generation part, green represents the network training part of DD, and red represents the network training part of tPD.

We illuminate the inference results and absolute error maps of four samples in Fig. 9. As expected, DD infers the sample phase while removing the imaging aberration phase, while the inference result of tPD includes both the sample phase and the aberration phase. Accordingly, the SSIM of DD is much higher than that of tPD. In DD, the hologram contains unwanted aberration information, but the GT only contains sample information, which means that the dataset implicitly contains both the prior for phase recovery and the prior for aberration removal. As for tPD, the prior for the network training is derived from numerical propagation, which allows both the sample information and the aberration information in the hologram to be recovered. It should be noted that the results of uPD also contain the unwanted aberration phase just like that of tPD.

3.5 Comparison of Experimental Data

We compare DD, tPD, CD, and uPD(tPDr) using experimental holograms with a defocus distance of 8.78 mm from an open-source dataset of Ref. 58. To match the defocus distance of the experimental hologram, we use ImageNet to generate corresponding datasets for the network training. Inference results of the standard phase object are shown in Fig. 10.

Overall, uPD and tPDr with multiple-cycle inferences have the best results, as seen from the neatly drawn peaks and valleys. It should be noted that due to the presence of redundant

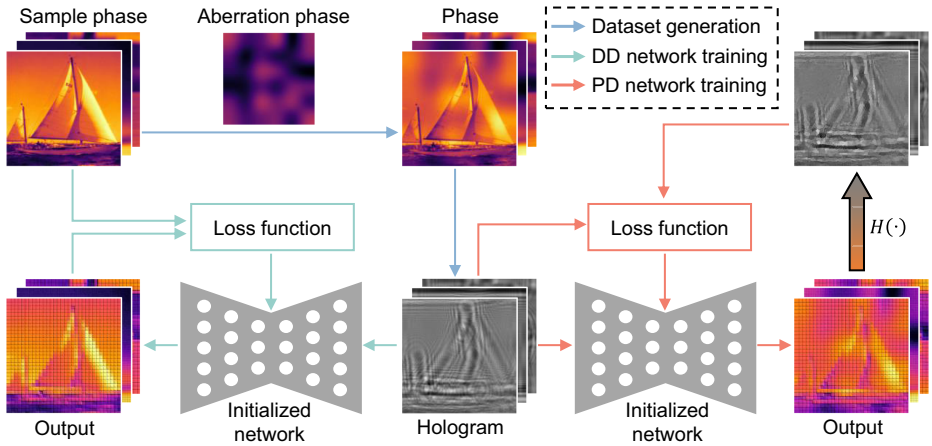

Fig. 8 Dataset generation and network training for the case of imaging aberration.

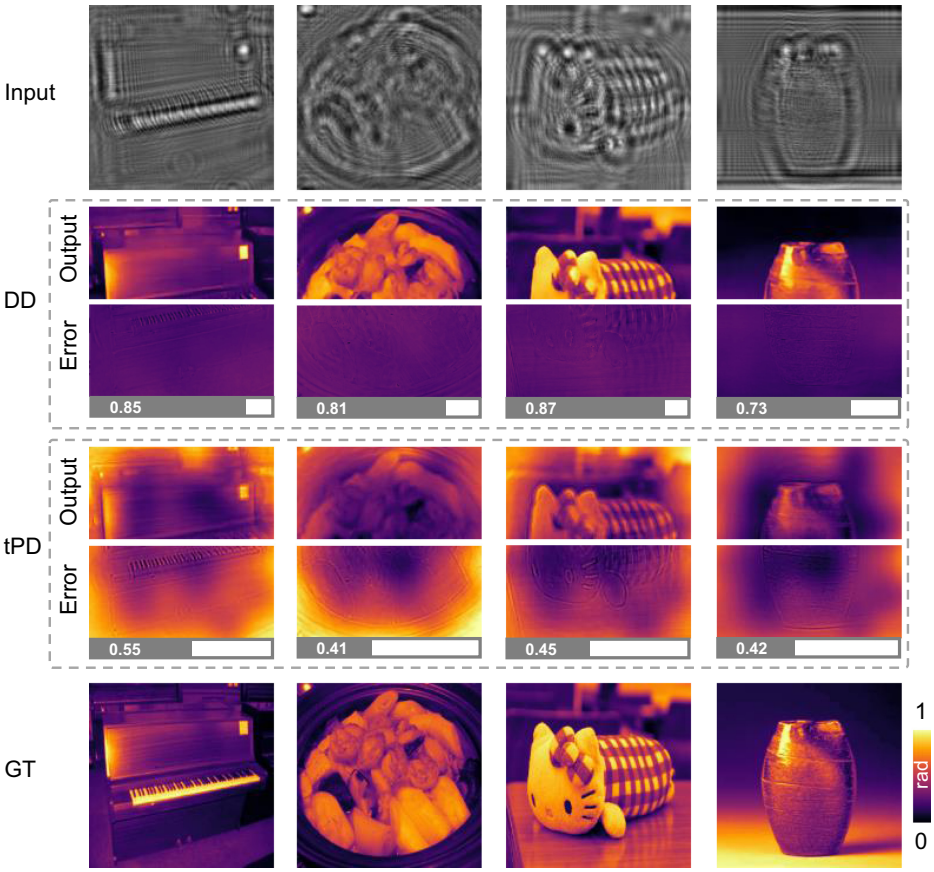

Fig. 9 Prior capacity test of DD and tPD.

diffraction fringes at the edge of the hologram [see the green box in Fig. 10(a)], unwanted fluctuations appear in the background of the uPD and tPDr inference results [see the green arrows in Fig. 10(a)]. Among the remaining one-time inference methods, the background fluctuations of the tPD results are larger [see the yellow arrows in Fig. 10(a)], while the detailed information of the DD results is weaker [see the yellow arrows in Fig. 10(b)]. As a combination of DD and tPD, CD better considers detailed and background information. It should be noted that as the training dataset further expands, the neural network's accuracy will increase accordingly. In addition, we also test tissue slices and get similar conclusions, as detailed in Fig. S2 in the Supplementary Material.

4 Conclusion

We introduced the principles of DD and PD strategies for deep learning phase recovery in the same context. On this basis, we compared the time consumption and accuracy of DD, uPD, tPD, and tPDr and found that uPD and tPDr achieve the highest

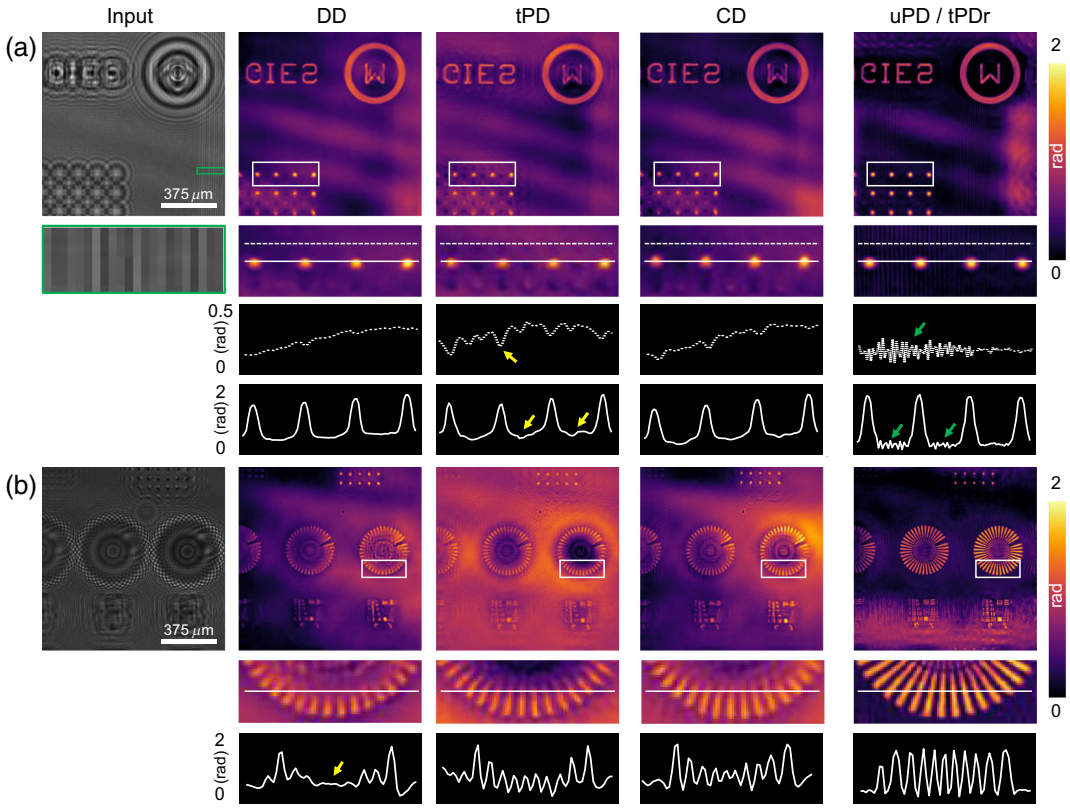

Fig. 10 Experimental tests of DD, tPD, CD, and uPD(tPDr). (a) Inference results of one field of view. (b) Inference results of another field of view.

accuracy with multiple inferences and tPD prefers the high-frequency detailed phase while DD favors the low-frequency background phase. Therefore, we proposed CD to balance high- and low-frequency information. Furthermore, we found that tPD generalizes better than DD for the case of inferring dense samples using neural networks trained on sparse samples. As for the case of inferring phase and amplitude simultaneously, we revealed the reason why DD is stronger than tPD, that is, the dataset for DD implicitly contains the mapping relationship from holograms to phase and amplitude while tPD may encounter situations where multiple network outputs phases and amplitudes correspond to a same hologram. To alleviate the ill-posedness of tPD, we proposed solutions by aperture constraints or multiple hologram inputs. In addition, we used the case of imaging aberration to demonstrate that DD can learn more about the prior implicit in the dataset whereas PD can only learn the prior in numerical propagation. Finally, we verified with experimental data that uPD and tPDr have the highest accuracy and that CD balances high- and low-frequency information better than DD and tPD.

We list some related papers with open-source code for readers to make further comparisons.[45–47,50]

5 Appendix A: Dataset Generation

Three publicly available image datasets (ImageNet, LFW, and MNIST) are used to generate phases and amplitudes, and then the corresponding holograms at a certain propagation distance are computed via numerical propagation. The training and testing datasets contain 10,000 and 100 data, respectively. The size of all data is set to 256×256. The propagation distance is set to 20 and 8.78 mm for the simulation comparisons and the experimental tests, respectively. In the code, we provide a hyperparameter "pad" to choose whether to use the way of "padding and cropping" to eliminate edge diffraction effects (see Fig. S3 in the Supplementary Material).

6 Appendix B: Network Implementation

The Adam optimizer with an initial learning rate of 0.001 is adopted to update the weights and biases. The Adam weight decay of uPD and tPDr is set to 0.001. The learning rate decreases to 0.95 of its current value every 5 or 10 epochs until it approaches 0.00001. The batch size of DD, tPD, and CD is set to 16. The neural network training epoch of DD and PD is set to 100. The inference cycles of uPD and tPDr are set to 10,000 and 1000, respectively. All the neural networks are based on PyTorch (2.0.0) with Python (3.8.18). All operators run on a compute server equipped with AMD Ryzen Threadripper PRO 3955WX and NVIDIA GeForce RTX 3090.

Disclosures

The authors declare no competing interests.

Code and Data Availability

The codes for DD, PD, and CD are publicly available at https://github.com/kqwang/DLPR.

Acknowledgment

The work was supported in part by the Research Grants Council of Hong Kong (Grant Nos. GRF 17201620, GRF 17200321, and RIF R7003-21).

References

1. K. Wang et al., "On the use of deep learning for phase recovery," *Light: Sci. Appl.* **13**(1), 4 (2024).
2. Y. Park, C. Depeursinge, and G. Popescu, "Quantitative phase imaging in biomedicine," *Nat. Photonics* **12**(10), 578–589 (2018).
3. R. K. Tyson and B. W. Frazier, *Principles of Adaptive Optics*, 5th ed., CRC Press, Boca Raton (2022).
4. J. Miao et al., "Extending the methodology of X-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens," *Nature* **400**(6742), 342–344 (1999).
5. R. Leach, Ed., *Optical Measurement of Surface Topography*, Springer, Berlin, Heidelberg (2011).
6. M. V. Klibanov, P. E. Sacks, and A. V. Tikhonravov, "The phase retrieval problem," *Inverse Problems* **11**(1), 1–28 (1995).
7. D. Gabor, "A new microscopic principle," *Nature* **161**, 777–778 (1948).
8. J. W. Goodman, *Introduction to Fourier Optics*, 4th ed., W.H. Freeman, New York (2017).
9. J. Hartmann, "Bermerkungen über den bau und die justierung von spektrographen," *Z. Instrumentenkd.* **20**, 47–58 (1900).
10. R. V. Shack and B. C. Platt, "Production and use of a lenticular Hartmann screen," *J. Opt. Soc. Am.* **61**, 656 (1971).
11. M. R. Teague, "Deterministic phase retrieval: a Green's function solution," *J. Opt. Soc. Am.* **73**(11), 1434–1441 (1983).
12. C. Zuo et al., "Transport of intensity equation: a tutorial," *Opt. Lasers Eng.* **135**, 106187 (2020).
13. G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica* **6**(8), 921–943 (2019).
14. K. Wang and E. Y. Lam, "Deep learning phase recovery: data-driven or physics-driven?" in *2024 Photonics and Electromagn. Res. Symp. (PIERS)*, pp. 1–4, IEEE (2024).
15. A. Sinha et al., "Lensless computational imaging through deep learning," *Optica* **4**(9), 1117–1125 (2017).
16. H. Wang, M. Lyu, and G. H. Situ, "eHoloNet: a learning-based end-to-end approach for in-line digital holographic reconstruction," *Opt. Express* **26**(18), 22603–22614 (2018).
17. M. J. Cherukara, Y. S. G. Nashed, and R. J. Harder, "Real-time coherent diffraction inversion using deep generative networks," *Sci. Rep.* **8**(1), 16520 (2018).
18. T. Nguyen et al., "Deep learning approach for Fourier ptychography microscopy," *Opt. Express* **26**(20), 26470–26484 (2018).
19. Z. Ren, Z. M. Xu, and E. Y. Lam, "End-to-end deep learning framework for digital holographic reconstruction," *Adv. Photonics* **1**(1), 016004 (2019).
20. L. Hu et al., "Deep learning assisted Shack–Hartmann wavefront sensor for direct wavefront detection," *Opt. Lett.* **45**(13), 3741–3744 (2020).
21. K. Wang et al., "Transport of intensity equation from a single intensity image via deep learning," *Opt. Lasers Eng.* **134**, 106233 (2020).
22. D. Pirone et al., "Speeding up reconstruction of 3D tomograms in holographic flow cytometry via deep learning," *Lab Chip* **22**(4), 793–804 (2022).
23. D. J. Chang et al., "Deep-learning electron diffractive imaging," *Phys. Rev. Lett.* **130**(1), 016101 (2023).
24. Y. Xue et al., "Reliable deep-learning-based phase imaging with uncertainty quantification," *Optica* **6**(5), 618–629 (2019).
25. X. Li et al., "Quantitative phase imaging via a cGAN network with dual intensity images captured under centrosymmetric illumination," *Opt. Lett.* **44**(11), 2879–2882 (2019).
26. K. Wang et al., "Y-Net: a one-to-two deep learning framework for digital holographic reconstruction," *Opt. Lett.* **44**(19), 4765–4768 (2019).
27. K. Wang et al., "Y4-Net: a deep learning solution to one-shot dual-wavelength digital holographic reconstruction," *Opt. Lett.* **45**(15), 4220–4223 (2020).
28. T. Zeng, H. K. H. So, and E. Y. Lam, "RedCap: residual encoder-decoder capsule network for holographic image reconstruction," *Opt. Express* **28**(4), 4876–4887 (2020).
29. L. Huang et al., "Holographic image reconstruction with phase recovery and autofocusing using recurrent neural networks," *ACS Photonics* **8**(6), 1763–1774 (2021).
30. H. Chen et al., "Fourier imager network (FIN): a deep neural network for hologram reconstruction with superior external generalization," *Light: Sci. Appl.* **11**(1), 254 (2022).
31. H. Chen et al., "eFIN: enhanced Fourier imager network for generalizable autofocusing and pixel super-resolution in holographic imaging," *IEEE J. Sel. Top. Quantum Electron.* **29**, 6800810 (2023).
32. X. Shu et al., "NAS-PRNet: neural architecture search generated phase retrieval net for off-axis quantitative phase imaging," arXiv:2210.14231 (2022).
33. Z. Ren, Z. M. Xu, and E. Y. Lam, "Learning-based nonparametric autofocusing for digital holography," *Optica* **5**(4), 337–344 (2018).
34. Z. Ren, H. K. H. So, and E. Y. Lam, "Fringe pattern improvement and super-resolution using deep learning in digital holography," *IEEE Trans. Ind. Inf.* **15**(11), 6179–6186 (2019).
35. K. Wang et al., "One-step robust deep learning phase unwrapping," *Opt. Express* **27**(10), 15100–15115 (2019).
36. Y. Zhu, C. H. Yeung, and E. Y. Lam, "Digital holographic imaging and classification of microplastics using deep transfer learning," *Appl. Opt.* **60**(4), A38 (2021).
37. Y. Zhu, C. H. Yeung, and E. Y. Lam, "Microplastic pollution monitoring with holographic classification and deep learning," *J. Phys.: Photonics* **3**(2), 024013 (2021).
38. L. Boominathan et al., "Phase retrieval for Fourier ptychography under varying amount of measurements," arXiv:1805.03593 (2018).
39. F. Wang et al., "Phase imaging with an untrained neural network," *Light: Sci. Appl.* **9**(1), 77 (2020).
40. X. Zhang, F. Wang, and G. H. Situ, "BlindNet: an untrained learning approach toward computational imaging with model uncertainty," *J. Phys. D: Appl. Phys.* **55**(3), 034001 (2022).
41. C. Bai et al., "Dual-wavelength in-line digital holography with untrained deep neural networks," *Photonics Res.* **9**(12), 2501 (2021).
42. D. Yang et al., "Dynamic coherent diffractive imaging with a physics-driven untrained learning method," *Opt. Express* **29**(20), 31426–31442 (2021).
43. D. Yang et al., "Coherent modulation imaging using a physics-driven neural network," *Opt. Express* **30**(20), 35647–35662 (2022).
44. L. Bouchama et al., "A physics-inspired deep learning framework for an efficient Fourier ptychographic microscopy reconstruction under low overlap conditions," *Sensors* **23**(15), 6829 (2023).
45. L. Huang et al., "Self-supervised learning of hologram reconstruction using physics consistency," *Nat. Mach. Intell.* **5**(8), 895–907 (2023).
46. O. Hoidn, A. A. Mishra, and A. Mehta, "Physics constrained unsupervised deep learning for rapid, high resolution scanning coherent diffraction reconstruction," *Sci. Rep.* **13**(1), 22789 (2023).
47. Y. Yao et al., "AutoPhaseNN: unsupervised physics-aware deep learning of 3D nanoscale Bragg coherent diffraction imaging," *NPJ Comput. Mater.* **8**(1), 124 (2022).
48. R. Li et al., "Physics-enhanced neural network for phase retrieval from two diffraction patterns," *Opt. Express* **30**(18), 32680–32692 (2022).
49. Y. Rivenson et al., "Phase recovery and holographic image reconstruction using deep learning in neural networks," *Light: Sci. Appl.* **7**(2), 17141 (2018).

50. M. Rogalski et al., "Physics-driven universal twin-image removal network for digital in-line holographic microscopy," *Opt. Express* **32**(1), 742 (2024).
51. I. Moon et al., "Noise-free quantitative phase imaging in Gabor holography with conditional generative adversarial network," *Opt. Express* **28**(18), 26284–26301 (2020).
52. L. Chen et al., "Image enhancement in lensless inline holographic microscope by inter-modality learning with denoising convolutional neural network," *Opt. Commun.* **484**, 126682 (2021).
53. Y. C. Wu et al., "Bright-field holography: cross-modality deep learning enables snapshot 3D imaging with bright-field contrast using a single hologram," *Light: Sci. Appl.* **8**(1), 25 (2019).
54. H. Luo et al., "Diffraction-Net: a robust single-shot holography for multi-distance lensless imaging," *Opt. Express* **30**(23), 41724–41740 (2022).
55. Z. Tian et al., "Lensless computational imaging with a hybrid framework of holographic propagation and deep learning," *Opt. Lett.* **47**(17), 4283 (2022).
56. K. Wang et al., "Deep learning spatial phase unwrapping: a comparative review," *Adv. Photonics Nexus* **1**(1), 014001 (2022).
57. Z.-Q. J. Xu et al., "Frequency principle: Fourier analysis sheds light on deep neural networks," *Commun. Comput. Phys.* **28**(5), 1746–1767 (2020).
58. Y. Gao and L. Cao, "Iterative projection meets sparsity regularization: towards practical single-shot quantitative phase imaging with in-line holography," *Light: Adv. Manuf.* **4**(1), 37–53 (2023).

Kaiqiang Wang is now a postdoctoral fellow of the Imaging Systems Laboratory (ISL) at the University of Hong Kong. He received his BS and PhD degrees from Northwestern Polytechnical University. His research interests include computational imaging and deep learning.

Edmund Y. Lam is now a professor in electrical and electronic engineering, professor in computer science (by courtesy), associate dean of the Graduate School, and director of ISL at the University of Hong Kong. He is a fellow of Optica, SPIE, IEEE, IS&T, IOP, and HKIE. His research interests include computational imaging algorithms, systems, and applications.