Supplemental Material

Deep Learning Coherent Diffractive Imaging

Dillan J. Chang, Colum M. O'Leary, Cong Su, Daniel Jacobs, Salman Kahn, Alex Zettl, Jim Ciston, Peter Ercius and Jianwei Miao

Supplemental Text

The stitching algorithm

Let $O_i(x,y)$ be a subset of the object phase function to update from the i^{th} reconstructed phase patch, $S_i(x,y)$, both of which are defined within a mask support where the CNN output is defined. The position of $O_i(x,y)$ on the overall object is defined by the scanning position from the ptychography experiment. The stitching algorithm performs gradient descent on the error function, $Err(O_i)$,

$$Err(O_i) = \frac{1}{2}\sum_x \sum_y \left[O_i(x,y) - \mu\big(O_i(x,y)\big) - S_i(x,y)\right]^2 \qquad (3)$$

where μ calculates the mean value of $O_i(x,y)$. By minimizing Eq. (3), we make the i^{th} phase object function be as close to S_i as possible after zero-centering, since an overall phase constant cannot be recovered from the CNN. Taking the gradient with respect to O_i, we get

$$\frac{\partial Err(O_i)}{\partial O_i} \propto [O_i - \mu(O_i) - S_i] \qquad (4)$$

and O_i is updated by,

$$O_{i,new} = O_i - \beta \frac{\partial Err(O_i)}{\partial O_i} \qquad (5)$$

where β is a constant between 0 and 1. In this work, we chose $\beta = 0.1$. Performing this update over all phase patches with $i = 1, \dots, N$ marks one iteration of the stitching algorithm, where N is total number of the phase patches. The algorithm usually quickly converges after several iterations.

Supplemental Figures

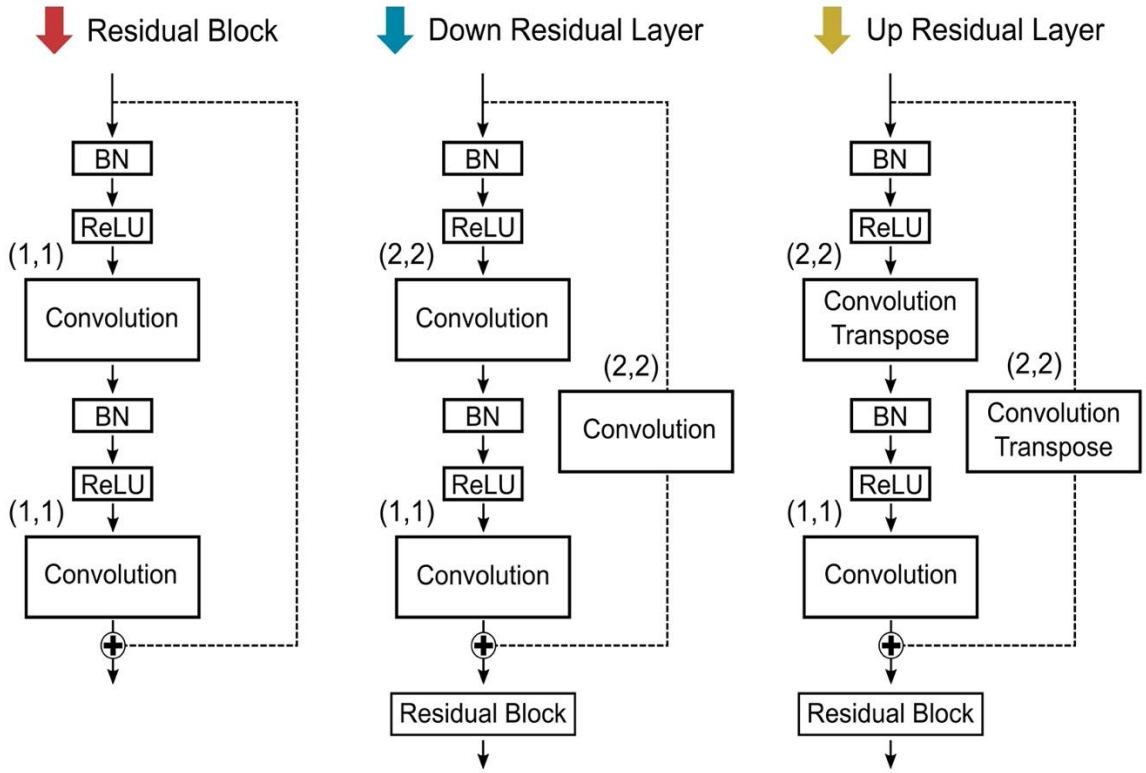

Fig. S1. Schematic of residual layers used in the CNN architecture. Different strides used in the convolution filters are shown in parentheses. All filters are size 3×3.

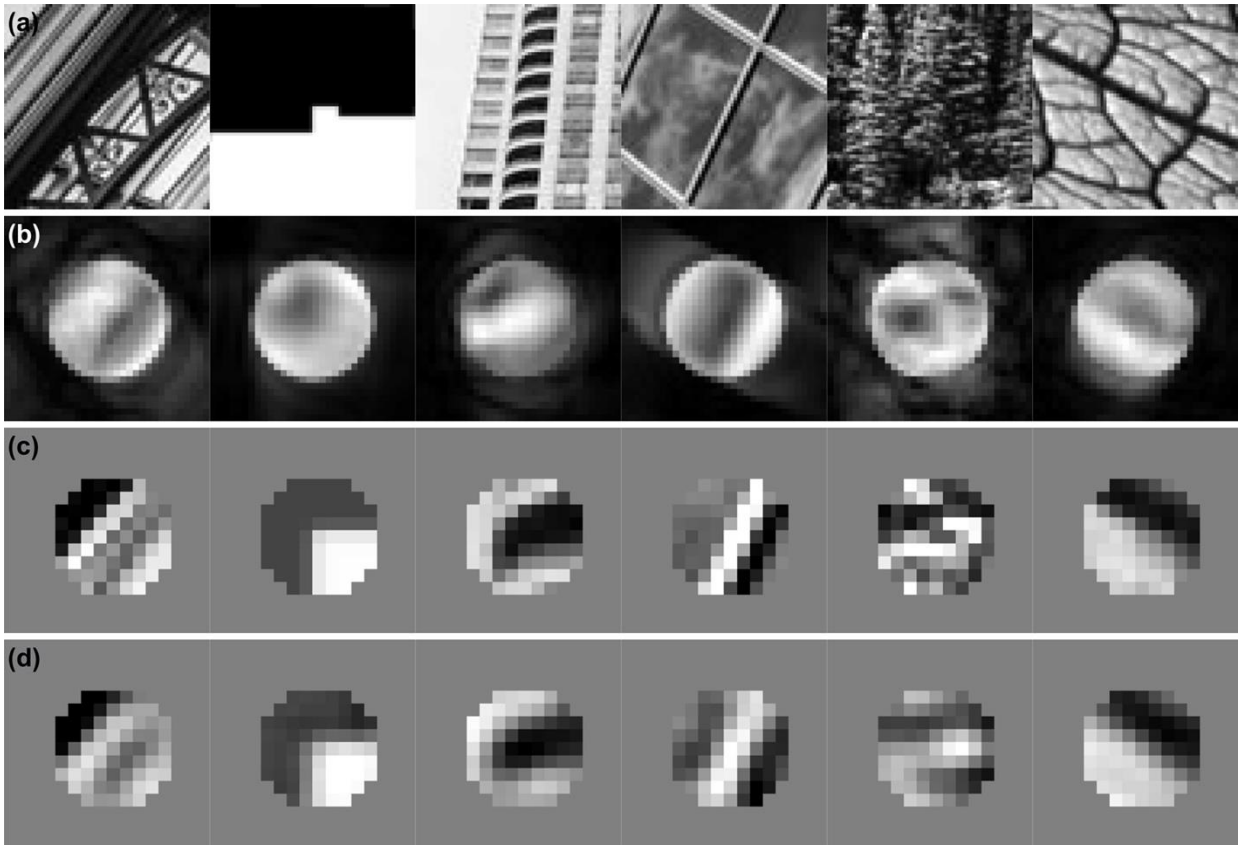

Fig. S2. Examples of augmented data generation and CNN performance. (a) Random images from the internet used as pure phase objects to train a CNN. (b) Diffraction patterns generated from the phase objects in (a). (c) Perfect phase patches within the illuminated areas. (d) Corresponding phase patches independently retrieved from the square root of the diffraction patterns by a trained CNN without any iteration.

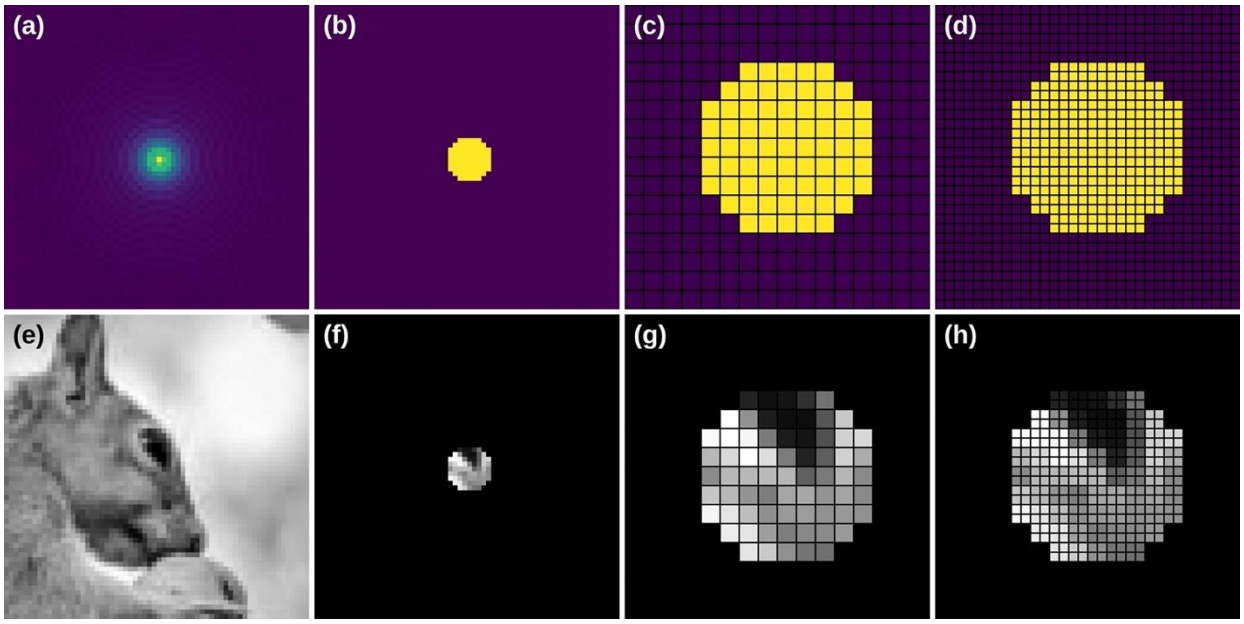

Fig S3. Procedure for generating real-space support and training data. (a) The magnitude of a probe function. (b) Real-space support based on the probe function. The phase object will only be predicted within this support, and any prediction outside of the support will not be used for stitching. (c) Cropped image of (b), removing non-trainable regions. Grids show individual pixels. (d) Oversampling (c) by a factor of 2. Cropping and oversampling in real space are optional steps to match the input/output dimensions of the CNN. (e) An example phase object. (f) A phase patch in the support predicted by the CNN. (g) Result of (f) after cropping non-trainable regions. (h) Result of (g) after oversampling the sample by a factor of 2. Any of (f), (g), or (h) can be used as training datasets for the CNN output given proper post-processing (padding and down-sampling) in the experiment.

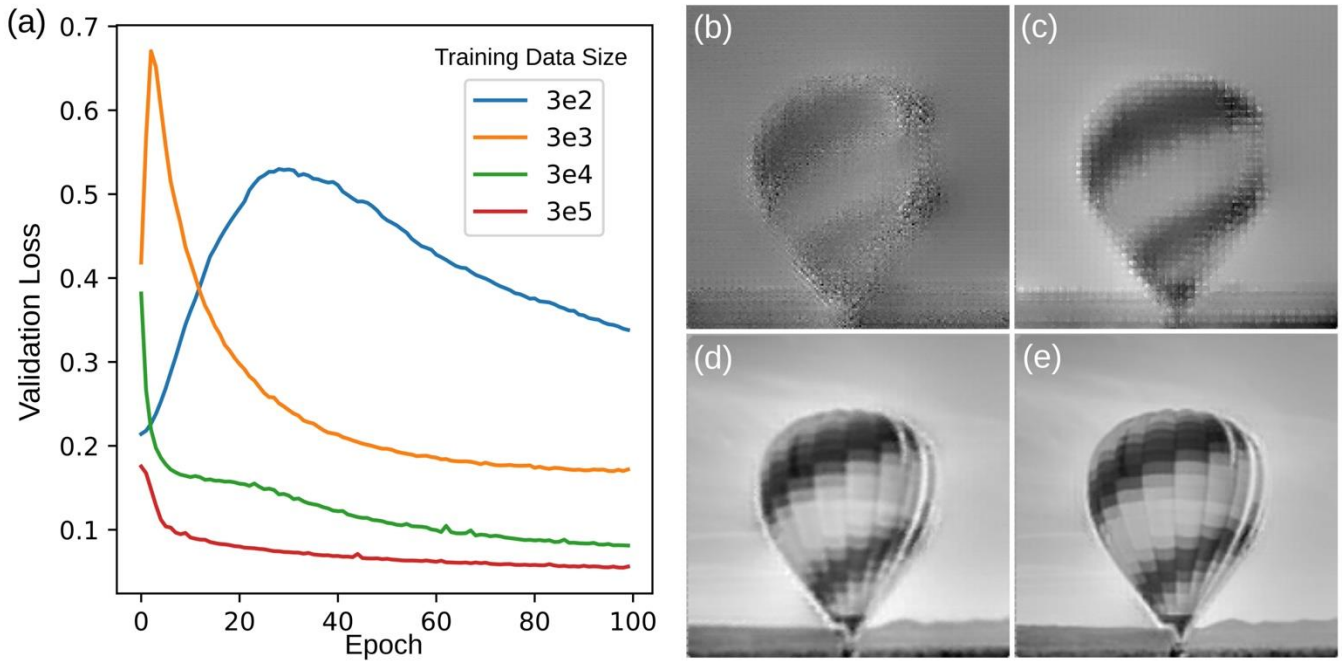

Fig. S4. Effect of training data size on CNN phase retrieval. (a) L1 validation loss when trained with various training data sizes. (b-e) Stitched phase reconstructions after 100 epochs of training with 3e2, 3e3, 3e4, and 3e5 training data sizes, respectively.

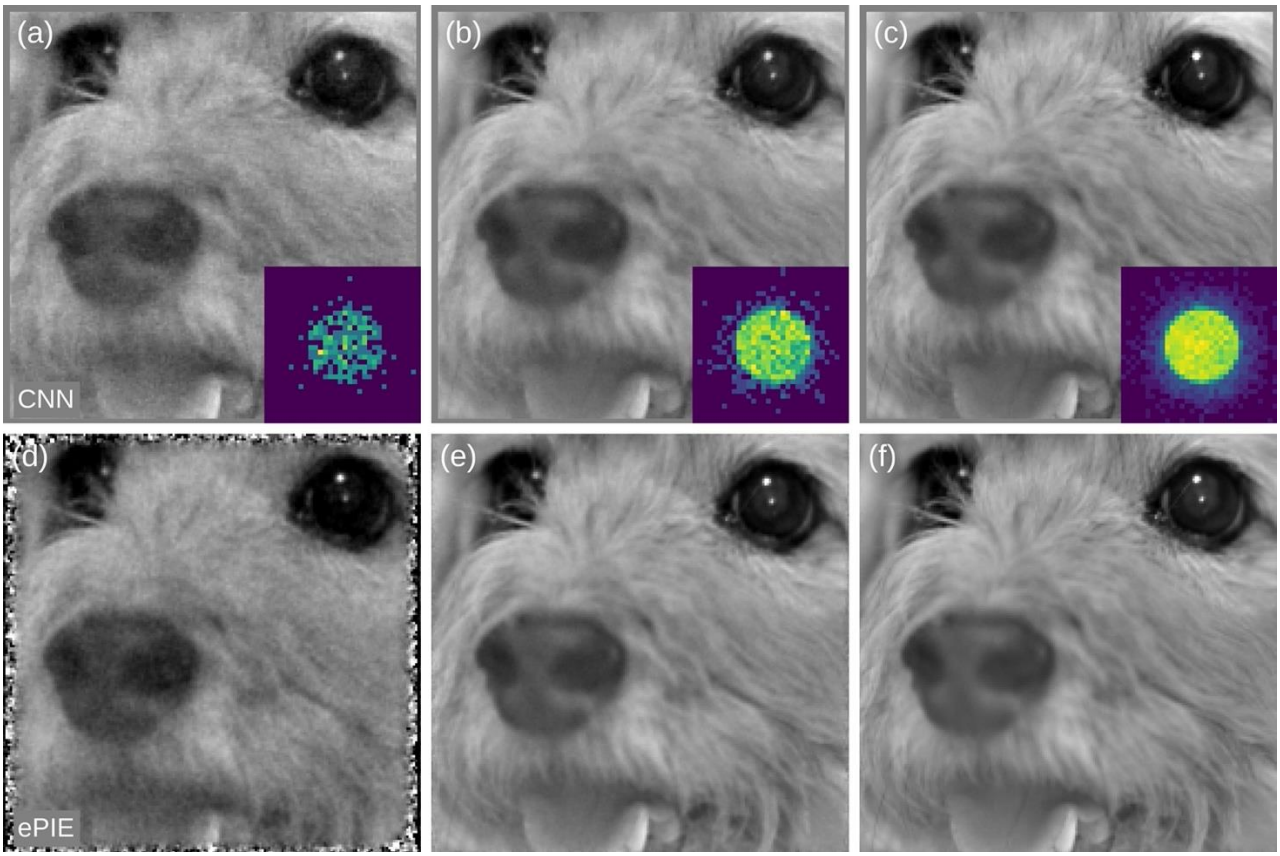

Fig. S5. Comparison between CNNs and ePIE under various shot noise conditions. (a-c) Stitched CNN phase reconstructions from diffraction patterns with 3e2, 3e3, and 3e4 events per pattern, respectively. Insets show representative diffraction patterns. (d-f) Phase reconstructions using ePIE from diffraction patterns with 3e2, 3e3, and 3e4 events per pattern, respectively.

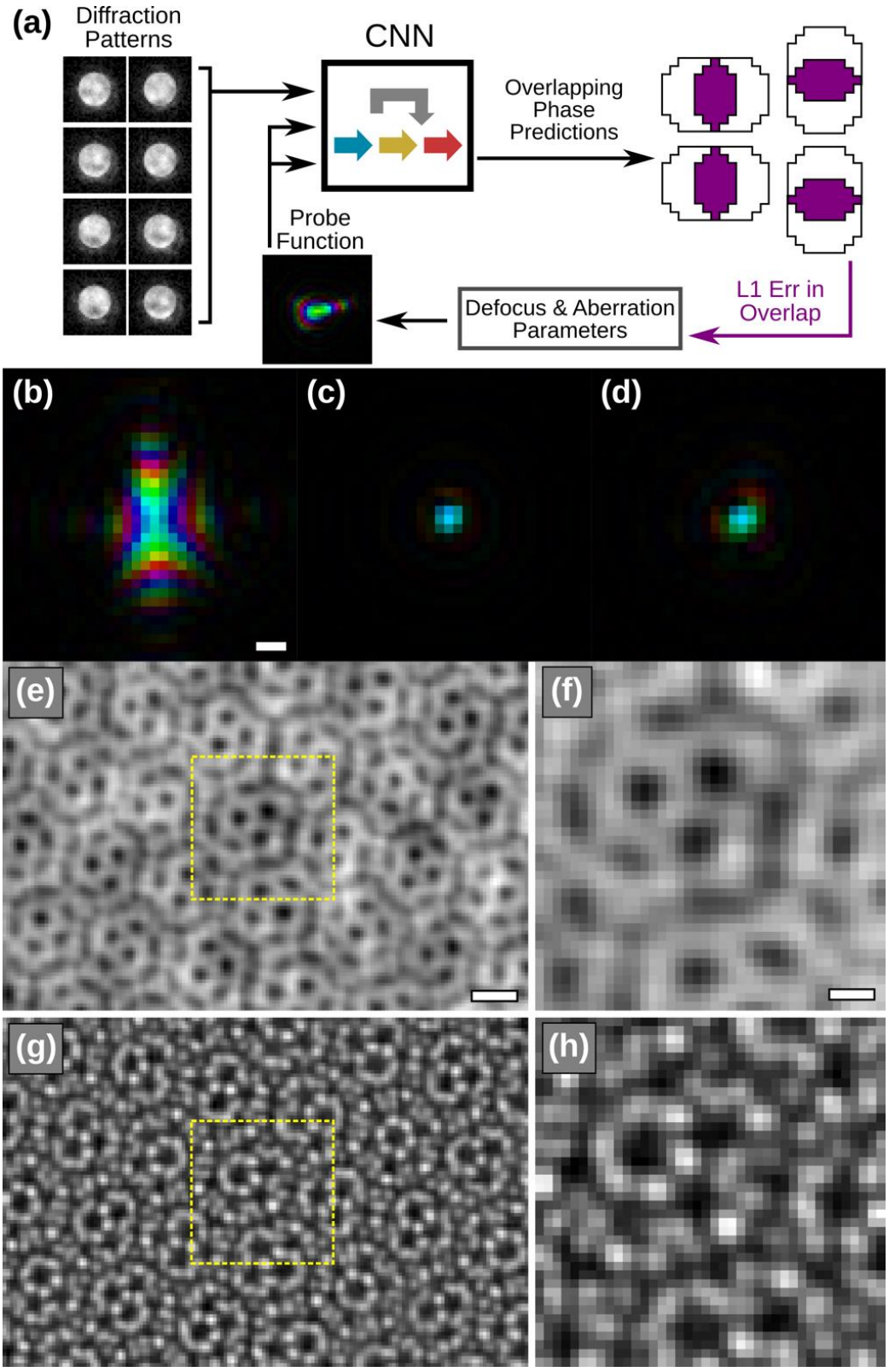
(a)
Diffraction Patterns
CNN
Overlapping Phase Predictions
Probe Function
Defocus & Aberration Parameters
L1 Err in Overlap
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Fig S6. Deep learning CDI with probe recovery. (a) Schematic of CNN-based probe recovery. An initial probe function is created by randomly choosing 7 defocus and aberration parameters. The CNN retrieves the phase patches from pairs of adjacent diffraction patterns with the initial probe function. The cumulative L1 error within the overlapping regions is used as the loss function to optimize the defocus and aberration parameters for the next iteration of the probe function. (b) Initial guess of the probe function for the hBN sample, where the brightness and hue represent the magnitude and phase of the probe function, respectively. (c) Probe function recovered by the CNN. (d) Probe function reconstructed by ePIE. (e) Phase image of the twisted hBN interface retrieved by the CNN with the initial probe function. (f) Magnified view of the dotted square in (e). (g) Phase image of the twisted hBN interface retrieved by the CNN with the probe function in (c). (h) Magnified view of the dotted square in (g). Scale bars, 1 Å (b); 4 Å (e); and 2 Å (f).

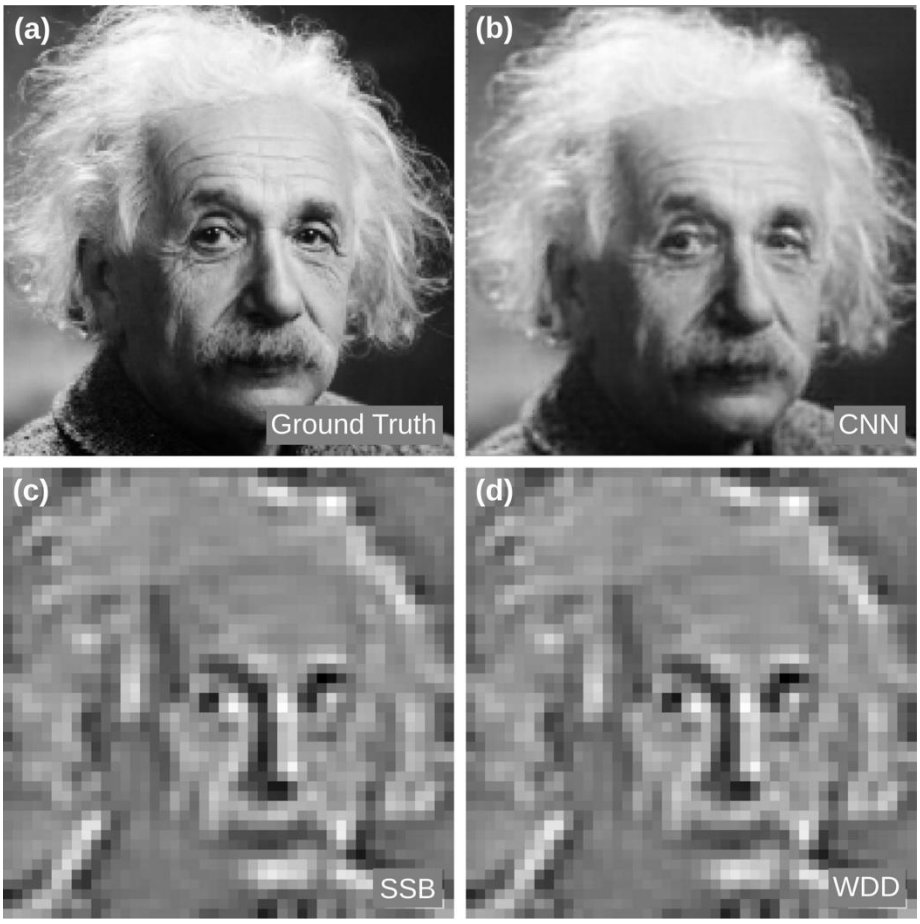

Fig. S7. Comparison of phase reconstructions using CNNs and non-iterative methods. Diffraction patterns were generated by a defocused probe of 6 pixels in diameter and a scanning step size of 4 pixels to show the difference in reconstruction quality. (a) Ground truth image simulated as a phase object. (b-d) Phase image reconstructions by CNNs, Single Side Band ptychography, and Wigner-Distribution Deconvolution, respectively.

Supplemental Tables

Table S1. Experimental parameters of raw data sets used for deep learning CDI.

Sample	Twisted hBN	Monolayer graphene [41]	Gold nanoparticle
Microscope	TEAM I	JEOL ARM200F	TEAM 0.5
Voltage (kV)	300	80	300
Convergence semi-angle (mrad)	17.1	31.5	17.1
Scanning step size (Å)	0.25	0.13	0.26
# probe positions	512×512	512×512	512×512
Dwell time per probe position (ms)	0.870	0.250	0.044
Pixelated detector	Gatan K3	JEOL 4DCanvas	4D Camera
# detector pixels (before binning)	512 × 512	66 × 66	576 × 576
Dose (e^{-} Å^{-2})	4.5×10^{6}	1.4×10^{6}	4.6×10^{4}

Table S2. Input parameters of experimental data sets used for deep learning CDI.

Sample	Twisted hBN	Monolayer graphene [41]	Gold nanoparticle
# detector pixels (after cropping + binning)	32	32	32
Maximum collection angle after cropping (mrad)	34.2	38.9	53.4
Angular pixel size (mrad/pixel)	1.07	1.21	1.67
Training sample size	250,000	250,000	250,000
Learning rate	1.e-4	1.e-4	1.e-4
# of trainable parameters	9,445,408	9,445,408	9,445,408
Epochs trained	100	100	100
Prediction time (ms per 1000 phase patches)	0.519	0.523	0.458

Table S3. Probe and diffraction pattern parameters used for CNN-based probe recovery.

Variable	Value
Binned detector dimensions (pixels)	32×32
Probe semi-convergence angle (pixels)	[3.2,12.8)
$C_{10}\alpha_{max}^2/\lambda$ (probe defocus)	[-10.0,10)
$C_{12a}\alpha_{max}^2/\lambda$ (probe twofold astigmatism)	[-10.0,10)
$C_{12b}\alpha_{max}^2/\lambda$ (probe twofold astigmatism)	[-10.0,10)
$C_{21a}\alpha_{max}^3/\lambda$ (probe coma)	[-10.0,10)
$C_{21b}\alpha_{max}^3/\lambda$ (probe coma)	[-10.0,10)
$C_{23a}\alpha_{max}^3/\lambda$ (probe threefold astigmatism)	[-10.0,10)
$C_{23b}\alpha_{max}^3/\lambda$ (probe threefold astigmatism)	[-10.0,10)
RMSE of phase in mask	[0.,0.75)
Events per diffraction pattern	[3000,10000)
CNN input dimensions	32×32×3
Training sample size*	1,500,000

*The training sample size is large because the trained CNN is universal and can recover both the probe function and the phase image from the diffraction patterns under different experimental conditions.