High-resolution limited-angle phase tomography of dense layered objects using deep neural networks

Alexandre Goy, Girish Rughoobur, Shuai Li, Kwabena Arthur, Akinunde I. Akinwande, and George Barbastathis

*3D Optics Laboratory, Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139; 2Microsystems Technology Laboratories, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139; and 3BioSystems and bioMechanics (BioSyM) Interdisciplinary Research Group, Singapore-MIT Alliance for Research and Technology (SMART) Centre, Singapore 117543, Singapore

We present a machine learning-based method for tomographic reconstruction of dense layered objects, with range of projection angles limited to $\pm 10^\circ$. Whereas previous approaches to phase tomography generally require 2 steps, first to retrieve phase projections from intensity projections and then to perform tomographic reconstruction on the retrieved phase projections, in our work a physics-informed preprocessor followed by a deep neural network (DNN) conduct the 3-dimensional reconstruction directly from the intensity projections. We demonstrate this single-step method experimentally in the visible optical domain on a scaled-up integrated circuit phantom. We show that even under conditions of highly attenuated photon fluxes a DNN trained only on synthetic data can be used to successfully reconstruct physical samples disjoint from the synthetic training set. Thus, the need for producing a large number of physical examples for training is ameliorated. The method is generally applicable to tomography with electromagnetic or other types of radiation at all bands.

Significance

We demonstrate that it is possible to use deep neural networks to produce tomographic reconstructions of dense layerd objects with small illumination angle as low as $10^\circ$. It is also shown that a DNN trained on synthetic data can generalize well to and produce reconstructions from experimental measurements. This work has application in the field of X-ray tomography for the inspection of integrated circuits and other materials studies.

Author contributions: A.I.A. and G.B. designed research; A.G. performed research; G.R., S.L., and K.A. contributed new reagents/analytic tools; A.G. analyzed data; and A.G., G.B., and G.V. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

1To whom correspondence may be addressed. Email: agoy@goyman.com.

First published September 16, 2019.
functions where the object class is sparse a priori known. Alternatively, if a database of representative objects is available, then these examples may be used to learn the optimal set of basis functions as a dictionary (28, 29).

Rapid recent developments in the field of machine learning, and deep neural networks (DNNs) in particular, have provided an additional set of tools and insights for inverse problems. It may be shown (31, 32) that recurrent or unfolded multistage DNN architectures are formally equivalent to the iterative solution to the inverse problem in Eq. 1 where the prior $\Phi$ need no longer be known or depend on sparsity; instead, examples guide the discovery of the prior through the DNN training process. Simpler learning architectures, where $g$ is fed to the DNN directly or after first passing through a preprocessor, have been used for retrieval of phase from intensity (33–36); 3D holographic reconstruction (37–39); superresolution photography (40–42) and microscopy (43); imaging through scatter (44–47); and imaging under extremely low light conditions in the 3 contexts computational ghost imaging (48), consumer-camera photography (49), and phase retrieval (50).

Multistage DNN architectures have been shown to yield high-quality reconstructions in numerous Radon tomography configurations (32, 40–50). Recently, Nguyen et al. (57) used the inverse Radon transform for optical tomography with a single-stage DNN intended to partially correct for the assumption of line integrals breaking down.

In this paper, we apply a Fourier-based beam propagation method (BPM) (58) as a preprocessor step immune from any Radon assumptions. The strongly scattering object is illuminated by a parallel beam under a limited angular range of $20^\circ$, i.e., $\pm 10^\circ$ from the reference axis. Unlike the earlier works on refractive index tomography referenced above, we do not perform a phase retrieval step; rather, the intensity measurements are preprocessed to produce directly an initial crude 3D guess of the object’s interior. This crude guess is then fed to our machine-learning algorithm. The preprocessing step is necessary because, even if we did convert intensity to phase, the results would not be interpretable as line integrals under our experimental conditions. Moreover, by merging phase retrieval and tomography into a single step, our algorithm becomes less sensitive to error accrual.

Large datasets, typically consisting of more than 5,000 examples, are generally required for DNN training. That is feasible in many cases through spatial light modulators (33, 57). However, in many cases of interest spatial light modulators have insufficient space-bandwidth product or are unavailable (e.g., in X-rays); and alternatives to generate physical specimens are expensive or restricted due to proprietary processes. Instead, our approach is to train the DNN on purely synthetic data with the rigorous BPM forward model and then use a physical test specimen (phantom) to test the reconstruction quality with well-calibrated ground truth in experiments.

We chose to design our phantom as emulating the 3D geometry of integrated circuits (ICs). These would normally be inspected with X-rays, so we scaled up the feature dimensions in the phantom for visible wavelengths. The advantage of this choice is that IC layouts provide strong geometrical priors, e.g., Manhattan geometries, and our phantom also exhibited large spatial gradients and refractive index contrast to strengthen scattering. Thus, our methodology is directly applicable to all cases of tomography at optical wavelengths, e.g., 3D-printed specimen characterization and identification and biological studies in cells and tissue with moderate scattering properties. In each case, testing the ground truth would require the fabrication or the accurate simulation of different phantoms meeting the corresponding priors.

There is also value in the study of emulating X-ray inspection of ICs at visible wavelengths, as extensive outsourcing by the IC industry has created a growing concern that the ICs delivered to the customer may not match the expected design and that malicious features may have been added (59). However, in our emulation the phase contrast of the features against the background and the Fresnel number are both higher than typical corresponding IC configurations even at soft X-ray wavelengths.

One advantage of deep learning for inverse problems is speed. Solving [1] separately for each instance of $g$ is computationally intensive, and training a DNN is even more so. This is because both operations are iterative, and the latter is run on large datasets. On the other hand, once the DNN has learned the inverse map from the preprocessed version of $g$ to $\hat{f}$, the computations are feedforward only. For example, the IC layout priors we exploit here could, in principle, also be learned by dictionaries—but, under strong scattering conditions, the latter would require iterative optimization of Eq. 1 with the forward operator $H$ itself consisting of an expensive computational procedure in each iteration. In our approach, the preprocessing performed prior to the DNN is the most time-consuming operation, and therefore we aim at simplifying the preprocessing step as much as we can, i.e., tolerate a crude approximation, and leave it to the DNN to correct it. In our case, the execution time is 51 s (out of which only 300 ms are taken by the DNN, with the rest being the preprocessor), while learning tomography (60, 61), which is based on a similar gradient descent algorithm, takes 212 s and yields inferior reconstructions (see Fig. 4 for this, and see Materials and Methods for the computing hardware implementation).

**Optical Experiment**

We prepared a series of 4 glass wafers with etched structures representing patterned layers from an actual IC design. A schematic cross-section of the sample is shown in Fig. 1A. The glass plates are held together and aligned in a custom-made holder. Immersion oil is added between the plates to minimize parasitic reflections and also tune the phase shift associated with the patterns. The pattern depth was measured to be 575 nm, yielding a phase shift of $-0.33\pi$ rad for the particular oil used. Note that the phase shift is negative as the refractive index of the oil is lower than the refractive index of the glass. Details about the sample preparation and phase-shift measurements are given in Materials and Methods. The particular patterns etched on the sample are shown in Fig. 1B.

The experimental apparatus is detailed in Fig. 2. A collimated monochromatic plane wave from a continuous wave (CW) laser is incident on the sample, which is mounted on a 2-axis rotation stage. The sample is imaged through a demagnifying telescope to increase the field of view. The detector (an EM-CCD camera) is defocused from the image plane to simulate free-space propagation in an X-ray experiment where no imaging system can be used. Further details are given in Materials and Methods.

The strength of diffraction effects can be quantified with the Fresnel number $F = a^2 / (\lambda d)$, where $\lambda$ is the wavelength, $d$ the propagation distance, and $a$ the characteristic feature size of the object. The smaller the Fresnel number is, the stronger the effects of diffraction. For the glass phantom considered here and a defocus of 58 mm, $F = 0.7$ for the smallest features and $F = 5.5$ for the largest. The diffraction pattern is digitized on the camera for different sample orientations. This series of measurements is then passed through a numerical algorithm, described in the next section, whose aim is to yield a first approximate reconstruction, hereafter referred to as the “approximant.”

**Computation of the Approximant**

As mentioned above, the task of the DNN is significantly facilitated if the raw measurements are preprocessed to give an approximation of the solution. We use a simple gradient descent...
B

Layer 1
Layer 2
Layer 3
Layer 4
Cover
0.5 mm thick
fused silica
Immersion oil
Layer 4
Layer 2
Layer 3
Layer 1
10 mm
Illumination
Δz
A

Layer 3
Layer 4
Layer 1
Layer 2

Fig. 1. (A) Sample cross-section. The depth of the etched patterns was measured (Materials and Methods) and the refractive index of the oil was controlled to achieve a known phase shift of 0.32 rad. Δz = 0.5 mm. (B) IC patterns used for each of the 4 layers. The white background represents the original wafer thickness and the black areas indicate where the wafer has been etched.

Light propagation through the object can be computed using the following split-step Fourier BPM. In this method each layer is modeled as a 2D complex mask \( f_l(x, y) = \exp[\alpha(x, y) + j\varphi(x, y)] \) and the space between 2 successive layers as Fresnel propagation through a homogeneous medium whose index of refraction equals the average refractive index of the sample, as

\[
\alpha = F^{-1} \left\{ F \left\{ u_{l-1} f_{l-1} \right\} (k_x, k_y) e^{-j(k_0 \sqrt{k_x^2 + k_y^2} \Delta z)} \right\}.
\]

Here, \( u_l(x, y) \) is the optical field at layer \( l \), \( F \) the Fourier transform, \( \Delta z \) the distance between layers, and \( k \) the wavenumber in the medium between layers. Each measurement is a collection of \( N_i \) intensity patterns \( g_i(x, y) \), with \( i = 1, \ldots, N_i \), captured on the detector for each orientation \( \theta \) of the sample. In this work, we assume that the sample is a pure phase object; i.e., \( \alpha(x, y) = 0 \). This assumption is valid for the glass phantom in the optical domain as well as for a sample composed of copper and silicon in the X-ray domain at 17 keV, where the phase contrast (real part of the refractive index contrast) is about 80 times larger than the absorption contrast (imaginary part of the refractive index contrast).

From the measurements, we produce an approximation \( \tilde{f} \) of the phase function \( \varphi_l(x, y) \) for each layer \( l \) in the sample. We use the steepest gradient descent method with a fixed number of iterations \( K = 8 \) to generate the approximant. In what follows, we represent the measurements (consisting of \( M \) real pixel values) by a \((M \times 1)\) column vector \( g \), and the discretized object (consisting of \( N \) real voxel values) by a \((N \times 1)\) column vector \( f \). We then define a cost function \( J \) to minimize, consisting simply of a data fidelity term

\[
J = \frac{1}{2} \sum_{i=1}^{N} \left\| H_i(f) - g_i \right\|_2^2,
\]

where \( H_i \) denotes the forward operator that maps the object function \( f \) to a prediction of the measurement \( H_i(f) \), for a particular orientation \( \theta \) of the sample. In the problem presented here, the optical field will first propagate through the sample \( L \) layers, each of thickness \( \Delta z \), and then in free space to the detector over a distance \( d \). The forward operator can thus be written as a cascade of Fresnel propagation operations and thin mask multiplications corresponding to the object layers, i.e., successive applications of Eq. 2 written in operator form

\[
H_i(f) = \left| u_{\text{det}} \right|^2
\]

\[
u_{\text{det}} = F_d \text{diag}[f_1] \cdots F_{\Delta z} \text{diag}[f_L] F_{\Delta z} \text{diag}[f_L] u_{\text{inc}, i}
\]

where \( f_i \) is the vector of object function values in layer \( i \), \( F_{\Delta z} \) the Fresnel propagation operator over distance \( \Delta z \), \( u_{\text{inc}, i} \) the incident field, \( u_{\text{det}} \) the field on the detector, and \( \text{diag}[v] \) the diagonal matrix with vector \( v \) on the diagonal. The gradient descent iterative update can be written as

\[
f^{(k+1)} = f^{(k)} - s(\nabla f^{(k)}, J)^T,
\]

where \( f^{(k)} \) is the object estimate at iteration \( k \), \( s \) the step size, and \( \nabla f^{(k)} \) the gradient of \( J \) with respect to \( f \) evaluated at \( f^{(k)} \). We then set \( f = f^{(K)} \) with \( K \) chosen in advance, starting from \( f^{(0)} = 0 \). The detailed derivation of the gradient for the particular

![Fig. 2. Experimental apparatus: spatial filter and beam expander. L1 is 10 ×, 0.25 numerical aperture objective; L2 is a 100-mm lens, with a 5-μm pinhole F1 in the focal plane; L3 is a 200-mm lens; and L4 is a 100-mm lens. Aperture A1 cuts the outer diffraction lobes of the beam. The sample is mounted on a 2-axis rotation stage rotating along the x and y axes. The sample middle plane is imaged using a telescope lens system with magnification 0.50 ×. The camera is defocused by a distance \( \Delta z = 58 \) mm from the image plane.](image-url)
model in Eqs. 4 and 5 is given after the conclusion in the Derivation of the Gradient section. In Fig. 3, we give an example of 1 experimental intensity measurement \((g_1)\) taken from the series of tomographic projections and the corresponding approximating \(f^0\) obtained from the whole series using Eq. 6.

**DNN Architecture and Training**

We use a DNN to map the approximating to the final reconstruction \(f\). The DNN is a convolutional neural network with a DenseNet architecture (62). The implementation is the same as the DNN used in ref. 45 except that the number of dense blocks was reduced to 3 in both the encoder and the decoder, as we empirically observed that using more dense blocks did not result in a significant improvement of the results. We produce a total of 5,500 synthetic sets of measurements obtained by simulating the optical apparatus with the beam propagation method in Eq. 2. The synthetic measurements were subject to simulated shot noise and read noise equivalent to the noise levels found in the experiment. The shot noise was accounted for by converting the simulated measurement pixel intensities \(I\) expressed in average photon count per pixel per integration period of the detector to integer numbers of photons \(N\) following Poisson statistics. The actual optical power on the camera was measured with a power meter and converted to an average photon flux per detector pixel. Read noise following Gaussian statistics was subsequently added. The parameters (variance and average) of the noise were measured from a series of dark frames from the camera taken with the same gain (EM gain of 1) and integration time (2 ms) as the experimental measurements.

From each set of measurements, we produce a multilayer approximant using the gradient descent in Eq. 6. The examples are split in a training set of 5,000 examples, a validation set of 450 examples, and a test set of 50 examples. Each set of measurements (1 example) comprises 22 views corresponding to different orientations of the sample. The DNN is then trained to map the approximant to the ground truth used for the simulation. Each layer of the sample is assigned to a different channel in the DNN. We use the negative Pearson correlation coefficient (NPCC = − PCC) as loss function and train in 20 epochs with a batch size of 16 examples. For 2 images \(A\) and \(B\), the PCC is defined as

\[
PCC(A, B) = \frac{\sum (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum (A_i - \bar{A})^2 \sum (B_i - \bar{B})^2}}.
\]

The PCC (and therefore the NPCC too) is agnostic to scale and offset; i.e., \(PCC(aA + b, cB + d) = PCC(A, B)\) for \(a, b, c, d \in \mathbb{R}\). As a consequence, the DNN, which is trained by minimizing the NPCC, may apply some offset and scaling to the reconstruction. These parameters are not easily predictable; however, for a given DNN they are constant once training is complete, which allows us to correct the reconstructions. Offset and scale are obtained by least-squares linear regression between the DNN output and the ground truth from the synthetic test set examples (not including the experimental example).

**Results**

The method described in the previous sections was applied to the glass phantom shown in Fig. 1B. The synthetic measurements were subject to Poisson noise resulting from \(10^7\) photon flux per detector pixel, equal to the experimental photon flux, and an additive Gaussian noise with a SD of 13 counts. For DNN training, we compared 2 sets of approximants, obtained with \(K = 1\) and \(K = 8\) with and without total variation (TV) regularization. In the case \(K = 1\), the regularization parameter \(\alpha\) was set to 0.1 (step size 0.1). We chose a smaller value of 0.04 for the case \(K = 8\) (step size 0.05) because the proximal operator corresponding to the regularizer is applied at each iteration and its effect tends to accumulate. In the case \(K = 8\), the particular choice for the number of iterations is an empirical trade-off between computation time and accuracy. The same optimization parameters (step size and number of iterations) were used to compute the approximant of the IC phantom, and the result for each layer is shown in Fig. 4 E–H for \(K = 1\) and Fig. 4 I–L for \(K = 8\). The DNN reconstruction results are summarized in Fig. 4 M–P (\(K = 1\)) and Fig. 4 Q–T (\(K = 8\)). The approximant and the DNN reconstruction represent the phase modulation imposed by each layer in the sample. The absolute phase carries no useful information; therefore we are free to offset the reconstructed phase by an arbitrary constant. In the DNN reconstructions in Fig. 4I–L, the IC patterns (where the phase shift actually occurs) are typically reconstructed with 0 phase due to the rectified linear units (which project all negative values to 0) at the output layer of the DNN. We reassign the phase of the pattern to the nominal phase of −0.33 rad so that it can be visually compared to the ground truths in Fig. 4 M–P. An alternate approach leading to very similar results is to assign a 0 phase to the background.

The DNN reconstructions can be compared to those obtained using learning tomography (LT), a previously demonstrated optical tomography technique (60, 61) based on proximal optimization (FISTA) (63) with TV regularization (64). The role of the TV regularizer is to favor piecewise constant solutions while preserving sharp edges, which is especially well suited for IC patterns. LT was initially designed for holographic measurements and was modified here to work on intensity measurements by computing the gradient for the data fidelity term in Eq. 3. The essential difference in the LT optimization is that a TV filter playing the role of a proximal operator is applied at each iteration on the current solution. The LT reconstructions for the experimental dataset are shown in Fig. 4 A–D. These particular reconstructions were obtained after 30 iterations of gradient descent, a step size of 0.05, a regularization parameter \(\alpha = 0.04\), and 20 iterations for the TV regularizer at each step. The computation time of the \(f^0\) approximant is 51 s for \(K = 8\) (no regularization) and 6 s for \(K = 1\), including 570 ms for the DNN vs. 212 s for LT on the same processor (see Materials and Methods for hardware details).

In Table 1, we summarize the values of the PCC, which we use to quantify the quality of the reconstructions. The values are given for reconstructions on the synthetic test set (50 examples) and also the reconstruction of the single experimental example. Because the reconstruction quality turns out to depend strongly on the particular layer, we display the value for each of the 4 layers separately. As may be expected, the values for the LT are higher (better reconstructions) than those for the approximant as LT was run for 30 iterations vs. 8 for the approximant and that the latter was not regularized. The DNN reconstructions appear to be the best according to the PCC metric, which shows that, even while starting from a poor approximation, the DNN was
Fig. 4. (A–D) Proximal gradient descent with TV regularization, $K = 30$ iterations, for each layer 1 to 4. (E–H) Approximants generated from the experimental measurements with $K = 1$. (I–L) Approximants generated from the experimental measurements with $K = 8$. (M–P) Reconstructions from the DNN of each approximant E–H, respectively. (Q–T) Reconstructions from the DNN of each approximant I–L, respectively. (U–X) Idealized ground truth obtained from the sample specifications for layers 1 to 4. Note that the color bar range covers more than the range of the data, so there is no saturation effect on the images.

able to outperform LT. Note that a direct comparison between the performance of the DNN on the synthetic data and that on the experimental example is not fair because the ground truth is not known in the experiment. The ground truth used for the experimental example is an idealization from the design parameters used to fabricate the sample. We also indicate the values for reconstructions based on the regularized approximant (using the same regularization parameters as in the LT algorithm).
In terms of PCC, there is no significant difference from the unregularized case.

The reconstructions based on regularized approximants are shown in Fig. 5. By comparing these images with the unregularized reconstructions shown in Fig. 4, and also by considering the value of the PCC in Table 1, we conclude that the regularization has little effect, especially on the experimental reconstructions. Moreover, the TV regularization may not operate as a favorable preconditioner for the DNN. The choice of the TV operator as a regularizer is arbitrary and only based on our assumption that the solution should be piecewise constant. In fact, because of the small angle range, the approximants for the different phantom layers are quite similar to each other and the regularization may cancel information that the DNN could use to discriminate between them. Layers 3 and 4 can be said to look visually the same method, to infer the refractive index \( \phi(x) \) of the object. The approximant obtained after 1 or several iterations of the steepest gradient method does not provide a reconstruction of sufficient quality for the purpose of IC integrity assessment. The DNN, however, exploits the strong priors contained in the object geometry and yields reconstructions that are significantly improved over the approximant. One of the main motivations for using DNN is indeed the speed of execution; therefore, we want to limit any unnecessary preprocessing. In fact, trying to improve the quality of the approximant by simply iterating more the gradient descent does not yield significant improvement.

One significant challenge for the method we demonstrated is to provide a proper training set for the DNN. In the case of ICs, the training set is simply given by the many real layouts that are available. For more generic objects, a problem needs first to be formulated to clearly define the class of object on which the training will be performed. This is, however, the case in general for problems that involves DNNs. What has been shown here is the compelling improvement that DNN can bring to a phase tomography problem when the class of object is known.

More generally, there is a trade-off between the specificity of the required priors and the “difficulty” of an inverse problem—measured as degree of ill-posedness, e.g., the ratio of largest to smallest eigenvalue in a linearized representation of the forward operator. The problem we addressed here is severely ill-posed due to the presence of strong scattering within the object and the limited range and number of angular measurements we allowed ourselves to collect. Therefore, the rather restricted nature of ICs as a class prior is justified; while, at the same time, our approach is addressing an indisputably important practical problem. Detailed determination of the relationship between the degree of ill-posedness and the complexity of the object class prior would be a worthwhile topic for future work.

Derivation of the Gradient. This derivation follows a path similar to the derivation given in ref. 61. We start from Eq. 3:

\[
J = \frac{1}{2} \sum_{i=1}^{N_x} \| H_i(f) - g_i \|^2
\]

where \( H_i(f) \) is the forward operator for the \( i \)-th angular measurement, \( g_i \) is the measured intensity, and \( f \) is the DNN reconstruction. The solution to this optimization problem is given by the least squares solution:

\[
\hat{f} = (H^T H + \lambda I)^{-1} H^T g
\]

where \( \lambda \) is the regularization parameter.

Table 1. PCC, expressed in percentage (i.e., \( \text{PCC} \times 100 \)), of the reconstructions in the test set with respect to the ground truths for the approximant (not regularized) and the DNN reconstructions, labeled “DNN,” obtained from the unregularized approximant

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>62 ± 7</td>
<td>48</td>
<td>99 ± 0.3</td>
<td>80</td>
<td>99 ± 0.4</td>
<td>72</td>
<td>91 ± 2</td>
<td>65</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>43 ± 5</td>
<td>22</td>
<td>97 ± 1</td>
<td>56</td>
<td>96 ± 1</td>
<td>45</td>
<td>79 ± 7</td>
<td>37</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>49 ± 9</td>
<td>41</td>
<td>99 ± 0.4</td>
<td>77</td>
<td>94 ± 5</td>
<td>76</td>
<td>89 ± 3</td>
<td>62</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>24 ± 7</td>
<td>7</td>
<td>95 ± 1</td>
<td>38</td>
<td>92 ± 2</td>
<td>42</td>
<td>76 ± 7</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>75 ± 63</td>
<td>63</td>
<td>100 ± 0.2</td>
<td>75</td>
<td>100 ± 0.1</td>
<td>76</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>57 ± 6.5</td>
<td>31</td>
<td>98 ± 0.7</td>
<td>44</td>
<td>99 ± 0.4</td>
<td>45</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>62 ± 6.5</td>
<td>52</td>
<td>99 ± 0.3</td>
<td>80</td>
<td>99 ± 0.3</td>
<td>79</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>41 ± 8.1</td>
<td>12</td>
<td>96 ± 0.8</td>
<td>48</td>
<td>98 ± 0.6</td>
<td>43</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

We show the 2 cases \( K = 1 \) and \( K = 8 \) for the approximant calculation. The LT solution is obtained with \( K = 30 \) and is indicated on the right. The values for the DNN trained with regularized approximants are labeled “DNN reg.” The uncertainty values indicated correspond to the SD over the 50 examples of the test set. For each case, the values for the synthetic (simulated) and experimental examples are indicated in separated columns “Simul.” and “Exp.”, respectively. No uncertainty is given for the experimental case as it contains only 1 example.
Fig. 5. (A–D) Approximants generated from the experimental measurements with \( K = 1 \) and TV regularization with \( \alpha = 0.1 \). (E–H) Approximants obtained with \( K = 8 \) and TV regularization with \( \alpha = 0.04 \). (I–L) Reconstructions from the DNN of each approximant A–D, respectively. (M–P) Reconstructions from the DNN of each approximant E–H, respectively.

The gradient of \( J \) is defined as

\[
\nabla f J = \begin{bmatrix} \frac{\partial J}{\partial \varphi_1}, \ldots, \frac{\partial J}{\partial \varphi_N}, \frac{\partial J}{\partial \alpha_1}, \ldots, \frac{\partial J}{\partial \alpha_N} \end{bmatrix}_{F(i)},
\]

where the object function is defined as \( f = \exp[j \varphi - \alpha] \), with \( \varphi \) representing the phase delay and \( \alpha \) the absorption. In what follows we denote the gradient by \( \nabla \) for notational simplicity. We take the gradient of Eq. 9 and, by linearity of the derivation operation and denoting \( H_i(f) \) by \( H_i = (H_1, \ldots, H_M)^T \), we get

\[
\nabla J = \frac{1}{2} \sum_{i=1}^{N_e} \left[ \nabla (H_i^T H_i) - 2 \nabla (H_i^T g_i) \right].
\]

The term \( \nabla (g_i^T g_i) \) is absent because measurements \( g_i \) do not depend on the estimate \( f \). Then, by the definition

\[
\nabla H = \begin{bmatrix} \frac{\partial H_1}{\partial \varphi_1}, \ldots, \frac{\partial H_1}{\partial \varphi_N}, \frac{\partial H_1}{\partial \alpha_1}, \ldots, \frac{\partial H_1}{\partial \alpha_N} \\
\vdots \\
\frac{\partial H_M}{\partial \varphi_1}, \ldots, \frac{\partial H_M}{\partial \varphi_N}, \frac{\partial H_M}{\partial \alpha_1}, \ldots, \frac{\partial H_M}{\partial \alpha_N} \end{bmatrix},
\]

we get

\[
\nabla J = \sum_{i=1}^{N_e} [H_i^T \nabla H_i - g_i^T \nabla H_i] \quad \text{[13]}
\]

\[
= \sum_{i=1}^{N_e} [r_i^T \nabla H_i]. \quad \text{[14]}
\]
where $r_i$ is the residual defined as $r_i = H_i - g_i$. Finally, we get the expression required in Eq. 6:

$$
(\nabla J)^2 = \sum_{i=1}^{N_e} (\nabla H_i)^2 r_i.
$$  \hspace{1cm} \text{(15)}

In Eq. 15, $(\nabla H_i)^2$ is a matrix of size $(2N \times M)$ that is too large to be computed directly. Instead, we use a routine, described below, to calculate the vector $(\nabla H_i)^2 r_i$ directly. We remind the reader of Eqs. 4 and 5 that describe the forward model where we drop the index $i$ to simplify the notation as the expression assumes the same form for each sample orientation:

$$
H = |u_{d \text{et}}|^2
$$  \hspace{1cm} \text{(16)}

$$
u_{d \text{et}} = F_d \text{diag}[f_i] \ldots F_{\Delta \text{z}} \text{diag}[f_2] F_{\Delta \text{z}} \text{diag}[f_1] u_{\text{inc}}.
$$  \hspace{1cm} \text{(17)}

This forward operator allows for a convenient computation of the gradient by using a backpropagation scheme. We first calculate the gradient of Eq. 16.

$$
\nabla H = \nabla |u_{d \text{et}}|^2
$$  \hspace{1cm} \text{(18)}

$$
= \nabla (\text{diag}[u_{d \text{et}}]^2) u_{d \text{et}}
$$  \hspace{1cm} \text{(19)}

$$
= \text{diag}[u_{d \text{et}}] \nabla u_{d \text{et}} + \text{diag}[u_{d \text{et}}] \nabla u_{d \text{et}}
$$  \hspace{1cm} \text{(20)}

$$
= 2R \{\text{diag}[u_{d \text{et}}] \nabla u_{d \text{et}}\},
$$  \hspace{1cm} \text{(21)}

where the asterisk represents the complex conjugate. Thus,

$$
(\nabla H)^{\dagger} \nu = 2R \{ (\nabla u_{d \text{et}})^{\dagger} \nu \},
$$  \hspace{1cm} \text{(22)}

where the dagger represents the Hermitian transpose and we have defined $\nu = \text{diag}[u_{d \text{et}}] \nu$. Because it is not practical to compute the matrix $(\nabla u_{d \text{et}})^{\dagger}$ due to its size, we use a recursive scheme to compute $(\nabla u_{d \text{et}})^{\dagger} \nu$ directly. For that, we rewrite Eq. 17 as a recursive relationship for the optical field $u_{d \text{et}}$ just after layer $l$:

$$
u_{d \text{et}} = F_{\Delta \text{z}} u_{d \text{et}},
$$  \hspace{1cm} \text{(25)}

The optical field $u$ is thus known everywhere for a given object function $f$. We then propagate the residual $\nu$ backward through the sample by using the same propagation relationships:

$$
u_{l+1} = F_{\Delta \text{z}}^{\dagger} \nu
$$  \hspace{1cm} \text{(26)}

$$
u_{l+1} = F_{\Delta \text{z}}^{\dagger} \text{diag}[f_{l+1}] \nu_{l+1}
$$  \hspace{1cm} \text{(27)}

Note that the Fresnel operator is unitary; i.e., $F^{\dagger} = F^{-1}$. We take the gradient of Eqs. 23–25:

$$
\nabla u_1 = \text{diag}[u_{\text{inc}}] \nabla f_1
$$  \hspace{1cm} \text{(28)}

$$
\nabla u_i = \text{diag}[F_{\Delta \text{z}} u_{l-1}] \nabla f_i + \text{diag}[f_i] F_{\Delta \text{z}} \nabla u_{l-1}
$$  \hspace{1cm} \text{(29)}

$$
\nabla u_{d \text{et}} = F_d \nabla u_{d \text{et}}.
$$  \hspace{1cm} \text{(30)}

We then take the Hermitian transpose and multiply by the residual; we get

$$
(\nabla u_1)^{\dagger} \nu_1 = (\nabla f_1)^{\dagger} \text{diag}[u_{\text{inc}}] \nu_1
$$  \hspace{1cm} \text{(31)}

$$
(\nabla u_i)^{\dagger} \nu_i = (\nabla f_i)^{\dagger} [\text{diag}[F_{\Delta \text{z}} u_{l-1}] \nu_i + (\nabla u_{l-1})^T F_{\Delta \text{z}}^{\dagger} \text{diag}[f_{l+1}] \nu_{l+1}]
$$  \hspace{1cm} \text{(32)}

$$
(\nabla u_{d \text{et}})^{\dagger} \nu = (\nabla u_{d \text{et}})^{\dagger} F_d \nu.
$$  \hspace{1cm} \text{(33)}

We simplify the equations above by making use of Eqs. 24 and 25,

$$
(\nabla u_1)^{\dagger} \nu_1 = (\nabla f_1)^{\dagger} \text{diag}[u_{\text{inc}}] \nu_1
$$  \hspace{1cm} \text{(34)}

$$
(\nabla u_i)^{\dagger} \nu_i = (\nabla f_i)^{\dagger} [\text{diag}[F_{\Delta \text{z}} u_{l-1}] \nu_i + (\nabla u_{l-1})^T F_{\Delta \text{z}}^{\dagger} \text{diag}[f_{l+1}] \nu_{l+1}]
$$  \hspace{1cm} \text{(35)}

$$
(\nabla u_{d \text{et}})^{\dagger} \nu = (\nabla u_{d \text{et}})^{\dagger} F_d \nu.
$$  \hspace{1cm} \text{(36)}

where we have used Eq. 24. Finally, according to Eq. 22, we obtain layer $l$ of $(\nabla H)^{\dagger} \nu$

$$
(\nabla H)^{\dagger} \nu_{l+1} = 2R \{ (\nabla u_{d \text{et}})^{\dagger} \nu_1 \},
$$  \hspace{1cm} \text{(40)}

where $\Re$ denotes the imaginary part.

**Materials and Methods**

The experimental apparatus is shown in Fig. 2A. The light source is a CW He-Ne laser at 632.8 nm that is spatially filtered, expanded, and collimated into a quasi-plane wave with an airy disk intensity profile of 33 nm in diameter. The sample is mounted on a 2-axis rotation stage rotating along the x and y axes. The sample middle plane is imaged using a demagnifying telephoto ($\times 0.50$) lens system to enhance the effect of diffraction and increase the field of view on the detector. The detector is an EM-CCD (QImaging Rolera EM-C2) with a $1,004 \times 1,002$ array of 8 x 8-um pixels. To simulate the diffraction occurring in an X-ray measurement, the detector is defocused by a distance $\Delta z = 58$ mm from the image plane, which corresponds to Fresnel numbers ranging from 0.7 to 5.5 for the different object features.

The sample corresponds to a $10^3 \times$ scale-up of a real IC design. The original IC comprises 13 layers, including the doped layers. We kept only layers 5 to 8 from the original design (relabeled here 1 to 4) shown in Fig. 2C that contain copper patterns that would induce a significant phase delay in the X-ray regime. The 4 glass plates corresponding to the IC layers were cut in a $500$-um-thick silica wafer and $575$-um deep patterns (measured after fabrication with a Bruker Dektak XT stylus profiometer) were obtained by wet etching. To control the phase contrast and reduce parasitic reflections between the layers, we used an immersion oil (Fig. 2B) with a refractive index $n_0 = 1.4005 \pm 0.0002$ at 25 $^\circ$C from Cargille-Sacher Laboratories. According to the manufacturer, the refractive index of the oil is $n_0 = 1.4005 \pm 0.0002$ at 632 nm and $20^\circ$C. The refractive index of fused silica is $n_2 = 1.457$ at 632 nm and 20$^\circ$C (65), which gives a contrast of $\Delta n = 0.0565 \pm 0.0005$. The corresponding phase shift for a single pattern is then $\Delta \theta = k \Delta n = 0.323 \pm 0.006$ rad.

The sample layers are fabricated on double-sided polished $150$-mm-diameter and $500$-um-thick fused silica wafers. A $1$-mm-thick positive tone resist (Megapost SPR700) is spin coated at 3,500 rpm on both sides of the wafer and soft baked at $95^\circ$C for $30$ min in a convection oven. The backside was also coated for protection from the forthcoming wet etch. Scaled versions of IC designs in GDSII format are then patterned directly using a maskless aligner (MLA150; Heidelberg Instruments) with a $405$-nm laser and developed using an alkaline developer (Shipley Micropos MDF CD-26) for $45$ s followed by a deionized (DI) water rinse and $N_2$ drying. A hard bake at $120^\circ$C for $30$ min is carried out to stabilize the patterned features. A short descum of $2$ min at $1,000$-W and $0.1$-Torr $O_2$ pressure in a barrel ash is also performed to remove any resist residue. The wafers are subsequently etched for $7$ min at a rate of $\sim 80$ nm/min in buffered oxide etch. The resist is then stripped from the wafer by a long ash (1 h) followed by a Piranha
clean (3:1 H$_2$SO$_4$:H$_2$O$_2$), DI water rinse, and N$_2$ drying. Finally, the wafers are diced into squares of 50 mm × 50 mm and cleaned again with Piranha, DI water rinse, and N$_2$ drying.

The computation of the approximant is performed with the MATLAB software on an Intel i9-7900X processor running at 3.3 GHz. The DNN training and testing are performed under Keras with Tensorflow backend running on an NVIDIA Titan Xp graphics processing unit.

ACKNOWLEDGMENTS. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA15-17-C-9113).