Supplementary Information: Learned Hardware-in-the-loop Phase Retrieval for Holographic Near-Eye Displays

PRANEETH CHAKRAVARTHULA, University of North Carolina at Chapel Hill
ETHAN TSENG, TARUN SRIVASTAVA, Princeton University
HENRY FUCHS, University of North Carolina at Chapel Hill
FELIX HEIDE, Princeton University

ACM Reference Format:
Praneeth Chakravarthula, Ethan Tseng, Tarun Srivastava, Henry Fuchs, and Felix Heide. 2020. Supplementary Information: Learned Hardware-in-the-loop Phase Retrieval for Holographic Near-Eye Displays. *ACM Trans. Graph.* 39, 6, Article 186 (December 2020), 26 pages. https://doi.org/10.1145/3414685.3417846

In this document we provide additional derivations and results in support of the primary text. All the holographic images reported are captured on our prototype display with phase holograms computed using the proposed learned hardware-in-the-loop approach.

1 FORWARD MODEL APPROXIMATIONS

Several wave propagation models can be derived from the diffraction integral as discussed in the main manuscript. In this section, we discuss various formulations of the diffraction integral that are relevant for near-eye display holography.

1.1 Angular Spectrum Propagation

As described in the main manuscript, the diffraction integral for calculating the wave propagation can be expressed as

$$E_I(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(\zeta,\eta)E_R(\zeta,\eta)\cdot g(x-\zeta,y-\eta)d\zeta d\eta, \tag{1}$$

where the kernel

$$g(\zeta,\eta) = \frac{1}{j\lambda}\frac{\mathbf{exp}\left[jk\sqrt{d^2+\zeta^2+\eta^2}\right]}{\sqrt{d^2+\zeta^2+\eta^2}}. \tag{2}$$

is the impulse response of free space propagation. Invoking the convolution theorem lets us express Eq. (1) as

$$E_I = (H\circ E_R) * g = \mathcal{F}^{-1}(\mathcal{F}[H\circ E_R]\circ\mathcal{F}[g]). \tag{3}$$

Authors' addresses: Praneeth Chakravarthula, cpk@cs.unc.edu, University of North Carolina at Chapel Hill, Chapel Hill, NC; Ethan Tseng, Tarun Srivastava, eftseng@princeton.edu, Princeton University, Princeton, NJ; Henry Fuchs, fuchs@cs.unc.edu, University of North Carolina at Chapel Hill, Chapel Hill, NC; Felix Heide, fheide@cs.princeton.edu, Princeton University, Princeton, NJ.

0730-0301/2020/12-ART186 $15.00
https://doi.org/10.1145/3414685.3417846

where $\circ$ is the Hadamard element-wise product and $\mathcal{F}$ is the Fourier transform operator. Assuming a plane wave illumination, computing the Fourier transform of the kernel g in the above equation results in

$$E_I = \mathcal{F}^{-1}\Big(\mathcal{F}[H] \circ G\Big). \tag{4}$$

where G is the ASM transfer function given by

$$G(f_x, f_y; z) = \begin{cases} \exp\left[j2\pi\frac{z}{\lambda}\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}\right] & , \quad \sqrt{f_x^2+f_y^2} < \frac{1}{\lambda} \\ 0 & , \quad \text{otherwise} \end{cases}. \tag{5}$$

This propagation model is called the *angular spectrum propagation* of the wave field. Following the above equation, the diffractive wave field from the SLM, if Fourier-analyzed across any plane, can be identified as plane waves traveling in different directions away from the hologram plane. Therefore, the field amplitude across any point can be calculated as the summation of contributions of these plane waves, taking into account the phase shifts undergone during the propagation [Goodman 2005]. The angular spectrum method (ASM) assumes *no approximations*. It is equivalent to the Rayleigh-Sommerfeld solution and yield identical predictions of the diffracted wave field [Shen and Wang 2006],

1.2 Fresnel and Fraunhofer Approximations

Following the diffraction integral as described in Eq. 1 of main manuscript, notice that the integral computes the Euclidean distance between the points on the hologram plane and the image plane where the wave is propagated. The Euclidean distance $\rho = \sqrt{(\zeta - x)^2 + (\eta - y)^2 + d^2}$ can be expressed as

$$\rho = d\sqrt{1 + \left[\frac{(\zeta - x)}{d}\right]^2 + \left[\frac{(\eta - y)}{d}\right]^2}. \tag{6}$$

Next, by binomial expansion, the above equation can be expressed as

$$\rho = d\Big[1 + \ell^2/2 - \ell^2/8 + \cdots\Big], \tag{7}$$

where $\ell^2 = [(\zeta - x)^2 + (\eta - y)^2]/d^2$.
Now, if the distance between the hologram and image plane d is sufficiently large compared to $(\zeta - x)$ and $(\eta - y)$, then the ρ in the exponential can be approximated by

$$\rho \approx d\left(1 + [(\zeta - x)/d]^2/2 + [(\eta - y)/d]^2/2\right). \tag{8}$$

This approximation simplifies calculating the diffractive wavefields at a sufficiently far distance, resulting in the widely adopted Fresnel propagation model. This can be further simplified when the wave propagates to much larger distances from the hologram aperture, resulting in a far-field Fraunhofer propagation of the wave field.

2 ADDITIONAL DISCUSSION ON LIGHT TRANSPORT DEVIATIONS

In this section, we present additional details on the deviation from *ideal coherent light transport* in real-world experimental setups.

2.1 SLM Discretization and Fill Factor

Unlike a continuous aperture assumed by scalar diffraction integral, an SLM is discretized into pixels. For a fill factor of 100% of the SLM, the pixel size equals the pixel pitch and the SLM would act as a continuous aperture, although that is often not the case. A real SLM can be characterized by the number of pixels N and M with the

pixel pitches $\Delta\zeta$ and $\Delta\eta$, and the fill factors $\alpha, \beta \in [0, 1]$ in the ζ and η directions respectively. The transmittance of such an SLM can be modeled as

$$t_{\text{SLM}} = \textbf{rect}\left(\frac{\zeta}{N\Delta\zeta}, \frac{\eta}{M\Delta\eta}\right)[t_{ap} + t_{ds}], \tag{9}$$

where

$$t_{ap}(\zeta, \eta) = \Big[\textbf{rect}\left(\frac{\zeta}{\alpha\Delta\zeta}, \frac{\eta}{\beta\Delta\eta}\right) * \textbf{comb}\left(\frac{\zeta}{\Delta\zeta}, \frac{\eta}{\Delta\eta}\right)\Big]\textbf{exp}(j\Phi(\zeta, \eta)) \tag{10}$$

is the transmission function of the *active pixel area* displaying the phase pattern of the computed phase-only hologram $H(\zeta, \eta) = \textbf{exp}(j\Phi(\zeta, \eta))$, and

$$\begin{aligned} t_{ds}(\zeta, \eta) = \Bigg\{\Bigg[&\textbf{rect}\left(\frac{\zeta}{\Delta\zeta}, \frac{\eta}{\Delta\eta}\right) - \textbf{rect}\left(\frac{\zeta}{\alpha\Delta\zeta}, \frac{\eta}{\beta\Delta\eta}\right)\Bigg] \\ &* \textbf{comb}\left(\frac{\zeta}{\Delta\zeta}, \frac{\eta}{\Delta\eta}\right)\Bigg\} A_{ds}(\zeta, \eta)\textbf{exp}(j\phi_{ds}(\zeta, \eta)) \end{aligned} \tag{11}$$

is the transmission of the *dead space area* of the SLM pixels, with $A_{ds}(\zeta, \eta)$ and $\phi(\zeta, \eta)$ denoting the amplitude andphase modulations of the dead space areas, respectively. The terms $\textbf{rect}\left(\frac{\zeta}{\alpha\Delta\zeta}, \frac{\eta}{\beta\Delta\eta}\right)$ represents the active area of a single pixel of the SLM, and the convolution with comb-function $\textbf{comb}\left(\frac{\zeta}{\Delta\zeta}, \frac{\eta}{\Delta\eta}\right)$ represents the periodic appearance of the pixels in ζ and η directions. The whole SLM is of size $N\Delta\zeta \times M\Delta\eta$ that is expressed by the function $\textbf{rect}\left(\frac{\zeta}{N\Delta\zeta}, \frac{\eta}{M\Delta\eta}\right)$. From Eq. (9), it can be seen that the SLM introduces an extra complex amplitude to the hologram, which typically shows up as a zero-order intensity overlay, significantly distorting the reconstructed image pattern.

3 PHASE OPTIMIZATION AND WIRTINGER DERIVATIVES

As discussed in the main manuscript, we aim for compensating errors in real holographic projections occurring due to aberrations in the hardware display. To this end, we model the real-world deviations from the ideal coherent light transport via a deep neural network that is trained with a large number of real measured holographic projections. Once trained, this deep neural network, which we call an *aberration approximator*, models the aberrations occurring in a real hardware holographic display with respect to the ideal propagated wave. We then compensate for these aberrations in the real display modeled by the aberration approximator, by optimizing for *aberration-compensating* holographic phase patterns. In this section, we discuss computing the Wirtinger derivatives for the optimization problem discussed in Section 4 of the main manuscript. Briefly, our forward model is as follows: For a given phase hologram $H(\Phi)$, we compute an ideal propagated wave field z using a band-limited angular spectrum propagation [Matsushima and Shimobaba 2009], as discussed above

$$z = \mathcal{P}(H(\Phi)) = \mathcal{F}^{-1}\big(\mathcal{F}[H] \circ G\big), \tag{12}$$

where $H(\Phi) = e^{j\Phi}$ is the complex phase hologram and G is the band-limited ASM transfer function. The intensity image of this ideal propagated wave ($I_{\text{ideal}} = |z|^2$) is passed to the aberration approximator ($\mathcal{D}$) to generate the intensity of the aberrated wave field

$$\widetilde{I} = \mathcal{D}(|z|^2), \tag{13}$$

resulting from various real-world deviations, as discussed in the main manuscript. Ideally, we want to compute hologram phase patterns that result in real hardware reconstructions as close to the target intensity as possible. In other words, we want the distance between the real-hardware display output $\mathcal{D}(|z|^2)$ produced by the hologram

with phase pattern Φ and the target image I to be zero. We pose the holographic phase retrieval problem as the following optimization problem

$$\begin{aligned}\Phi_{\text{opt}} &= \min_{\Phi} \quad f(\mathcal{D}(|z|^2), I),\\ &= \min_{\Phi} \quad f(\mathcal{D}(|\mathcal{F}^{-1}\big(\mathcal{F}[H] \circ G\big)|^2), I),\\ &= \min_{\Phi} \quad \underbrace{f(\widetilde{I}, I)}_{Err(\Phi)},\end{aligned} \tag{14}$$

where f is a penalty function to compute the error between the target and reconstructed images. We build on the Wirtinger holography framework to solve the above optimization problem using first-order optimization methods. We briefly discuss here the Wirtinger gradients for our proposed optimization method.

3.1 Wirtinger gradients

In order to update the hologram phase patterns using a gradient descent optimization technique, we require the gradient of the error function in Eq. (14) with respect to the phase pattern Φ. This can be calculated applying the chain rule to Eq. (12) and (14) as

$$\frac{dErr}{d\Phi} = \underbrace{\frac{df}{dz}}_{A}\underbrace{\frac{dz}{dH}}_{B}\underbrace{\frac{dH}{d\Phi}}_{C}. \tag{15}$$

Notice that Part-A of the Eq. 15 requires us to compute the derivative of the scalar real-valued error f with respect to the complex diffractive wave field z. As the gradient of a scalar valued function with respect to a complex valued variable is zero or not defined, we approximate the partial of the scalar function of the complex vector, to overcome the undefined gradient [Chakravarthula et al. 2019] as

$$d(Err) = df(z) = Re\big\langle \nabla f, dz \big\rangle, \tag{16}$$

where Re denotes the real part of a complex number and $\langle ., . \rangle$ denotes the inner product of two vectors. Note that the above definition is not the exact gradient but only an approximate definition to use with any first-order optimization techniques. The value of ∇f in the above definition is obtained using the complex Wirtinger derivatives

$$\nabla f(z) = 2\nabla_{\bar{z}} f, \tag{17}$$

which can be further simplified by applying chain rule to Part-A of Eq. 15 as

$$\nabla_{\bar{z}} f = \Big[\frac{df}{d(\mathcal{D}(|z|^2))}\frac{d(\mathcal{D}(|z|^2))}{d(|z|^2)}\Big] \circ 2\nabla_{\bar{z}}(|z|^2). \tag{18}$$

Observe that the first part of the above Eq. 18 are partials of the scalar error function with respect to the aberrated image, and the partial of the aberrated image from the *aberration approximator* network with respect to the image from *ideal-propagation*. The gradients to both can be computed analytically using multivariate calculus, or obtained from the auto-differentiation functionality of the existing deep neural network frameworks. The second part of the Eq. 18 can be reduced to

$$2\nabla_{\bar{z}}(|z|^2) = 2\nabla_{\bar{z}}(z\bar{z}) = 2z. \tag{19}$$

Therefore, the Part-A of the gradient Eq. (15) is evaluated as follows

$$\frac{df}{dz} = \nabla f = \Big[\frac{df}{d(\mathcal{D}(|z|^2))}\frac{d(\mathcal{D}(|z|^2))}{d(|z|^2)}\Big] \circ 2z. \tag{20}$$

As discussed in Section 1, the ideal wave field at the destination plane can be obtained using an angular spectrum propagation method as $z = \mathcal{F}^{-1}\Big(\mathcal{F}[H] \circ G\Big)$, and the image is computed as $|z|^2$. Using this model, computing Part-B of the Eq. (15) using the Part-A gradient computed in Eq. (20) yields:

$$\begin{aligned} d(Err(H)) &= Re\Big\langle \nabla f, dz \Big\rangle, \\ &= Re\Big\langle \nabla f, d(F^\dagger(G)(FH)) \Big\rangle, \\ &= Re\Big\langle \nabla f, F^\dagger G F dH \Big\rangle, \\ &= Re\Big\langle F^\dagger G^\dagger F \nabla f, dH \Big\rangle, \\ &= Re\Big\langle F^\dagger G^* F \nabla f, dH \Big\rangle. \end{aligned} \tag{21}$$

Finally evaluating Part-C with the complex amplitude on the hologram plane as $H = e^{j\Phi}$, we derive the definition of gradient of the error function with respect to the phase Φ as follows:

$$\begin{aligned} d(Err(\Phi)) &= Re\Big\langle F^\dagger G^* F \nabla f, d(e^{j\phi}) \Big\rangle, \\ &= Re\Big\langle -je^{-j\phi} F^\dagger G^* F \nabla f, d(\phi) \Big\rangle. \end{aligned} \tag{22}$$

Since the phase Φ is real valued, the above inner-product definition can be read as:

$$d(Err(\Phi)) = \Big\langle Re(-je^{-j\phi} F^\dagger G^* F \nabla f), d(\phi) \Big\rangle,$$

$$\nabla Err(\Phi) = Re(-je^{-j\phi} F^\dagger H^* F \nabla f). \tag{23}$$

With the above gradient in hand, we optimize for the aberration-compensating phase patterns using standard first-order stochastic gradient descent solvers.

4 WIRTINGER DERIVATIVES IN MACHINE LEARNING LIBRARIES

Popular machine learning libraries such as Tensorflow and PyTorch now support complex valued variables and functions, and include built-in automatic differentiation capability, which was unavailable until recently. We believe that this will greatly facilitate the use of stochastic gradient descent (SGD) methods with complex valued functions for optimization and machine learning, where the user need not specifically derive the complex gradients. In this section, we discuss the complex Wirtinger derivatives and the implementation of automatic differentiation using chain rule in machine learning libraries.

A complex valued function of complex variables which is complex differentiable at every point in its domain is called a holomorphic function. For a given complex function of a complex variable, for instance

$$f : z \in \mathbb{C} \mapsto f(z) \in \mathbb{C}, \tag{24}$$

its derivatives can be defined as follows

$$f'(z_0) = \frac{df}{dz}\Big|_{z_0} = \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0}. \tag{25}$$

A given complex function can be decomposed into two real functions, each depending on two real variables, say x and y, which are the real and imaginary parts of the complex variable z. Mathematically, this can be represented as

$$f(z) = f(x + jy) \equiv u(x, y) + jv(x, y); z = x + jy. \tag{26}$$

It can be shown that for the above function $f(z)$ to be holomorphic, the corresponding component functions $u(x, y)$ and $v(x, y)$ need to satisfy the Cauchy-Riemann conditions [Remmert 2012] defined as follows:

$$\frac{\partial u(x, y)}{\partial x} = \frac{\partial v(x, y)}{\partial y} \tag{27}$$

and

$$\frac{\partial v(x,y)}{\partial x} = -\frac{\partial u(x,y)}{\partial y}. \tag{28}$$

This means that if $f : \mathbb{C} \mapsto \mathbb{C}$ is a function which is differentiable when regarded as a function on $\mathbb{R}^2$, then f is complex differentiable if and only if the Cauchy-Riemann equations hold. Note that u and v, as defined above, are *real-differentiable* functions of two real variables and $u + iv$ is a (complex-valued) real-differentiable function. However, $u + iv$ is *complex-differentiable* if and only if the Cauchy-Riemann equations hold. This insight leads to Wirtinger derivatives of a (complex) function $f(z)$ of a complex variable $z = x + jy$ defined as the following linear partial differential operators of the first order:

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right) \tag{29}$$

and

$$\frac{\partial f}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right). \tag{30}$$

We refer the reader to Chakravarthula et al. [2019; 2020] for a detailed discussion.

4.1 Chain Rule

With the above defined Wirtinger derivatives, the gradients of compound functions can be computed using chain rule. If $f, g \in C(\Omega)$, and $g(\Omega) \subseteq \Omega$, then the derivative of the function $f \circ g$ can be computed as:

$$\frac{\partial}{\partial z}(f \circ g) = \left(\frac{\partial f}{\partial z} \circ g\right)\frac{\partial g}{\partial z} + \left(\frac{\partial f}{\partial \bar{z}} \circ g\right)\frac{\partial \bar{g}}{\partial z} \tag{31}$$

and

$$\frac{\partial}{\partial \bar{z}}(f \circ g) = \left(\frac{\partial f}{\partial z} \circ g\right)\frac{\partial g}{\partial \bar{z}} + \left(\frac{\partial f}{\partial \bar{z}} \circ g\right)\frac{\partial \bar{g}}{\partial \bar{z}}. \tag{32}$$

While the above definition can be used to compute the complex gradients, such as in Tensorflow's library, one can also formulate the partial derivatives as forming the components of the Jacobian matrix. The derivative then can be expressed as a matrix representation of the following relation:

$$\mathbf{D}(f \circ g) = (\mathbf{D}f \circ g)\mathbf{D}g \tag{33}$$

where the Jaconbian on f, $\mathbf{D}f$, is defined as:

$$\mathbf{D}f = \begin{pmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \bar{z}} \\ \frac{\partial \bar{f}}{\partial z} & \frac{\partial \bar{f}}{\partial \bar{z}} \end{pmatrix}. \tag{34}$$

Now, for a function $f(x, y) = (u(x, y), v(x, y))$, we can express the above Equation (34) in the frame of $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ as the matrix

$$\mathbf{J}f = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} \tag{35}$$

along with the base change matrix, to change the basis to $\frac{\partial}{\partial z}$ and $\frac{\partial}{\partial \bar{z}}$, given by

$$\mathbf{P} = \frac{1}{2}\begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}. \tag{36}$$

Following the Wirtinger derivatives defined in Equations (29) and (30), the Jacobian operation $\mathbf{D}f$ in the $\frac{\partial}{\partial z}$, $\frac{\partial}{\partial \bar{z}}$ frame can now be expressed as $\mathbf{P}\ \mathbf{J}f\ \mathbf{P}^{-1}$. This formulation of complex gradients is used in the PyTorch

implementation of auto-gradients [1]. For non-holomorphic functions, whose gradients are either zero or not defined [Chakravarthula et al. 2019], both machine learning libraries give a *descent direction*.

5 ADDITIONAL PROTOTYPE DETAILS

Our hardware prototype is similar to the one demonstrated by Chakravarthula et al. [2019]. Our bench-top prototype is built using cage system mounting and the optics are adjusted for an image plane distance of about 200 mm from the SLM. We use red, green, and blue single-mode fiber lasers that are controlled by a Thorlabs LDC205C Laser Diode Controller, sequentially changing the three colors. The exposure of the imaging camera and the laser intensity are adjusted once based on the laser power. All capture settings are kept constant for experiments. The holographic image is directly mapped on the camera sensor. We use an additional phase ramp over the holograms to shift the image away from the zero-order undiffracted light. However, this makes the conjugate (ghost) images apparent. We filter both the zero-order and the conjugate images using an iris in the intermediate plane. Although this causes additional artifacts due to the SLM limitations, we are able to correct for those aberrations using our hardware-in-the-loop phase retrieval method. The improved contrast due to the zero-order elimination can be seen in Figure 1.

We will publish setup and optical design files to facilitate reproducibility.

[1] https://pytorch.org/docs/stable/notes/autograd.html

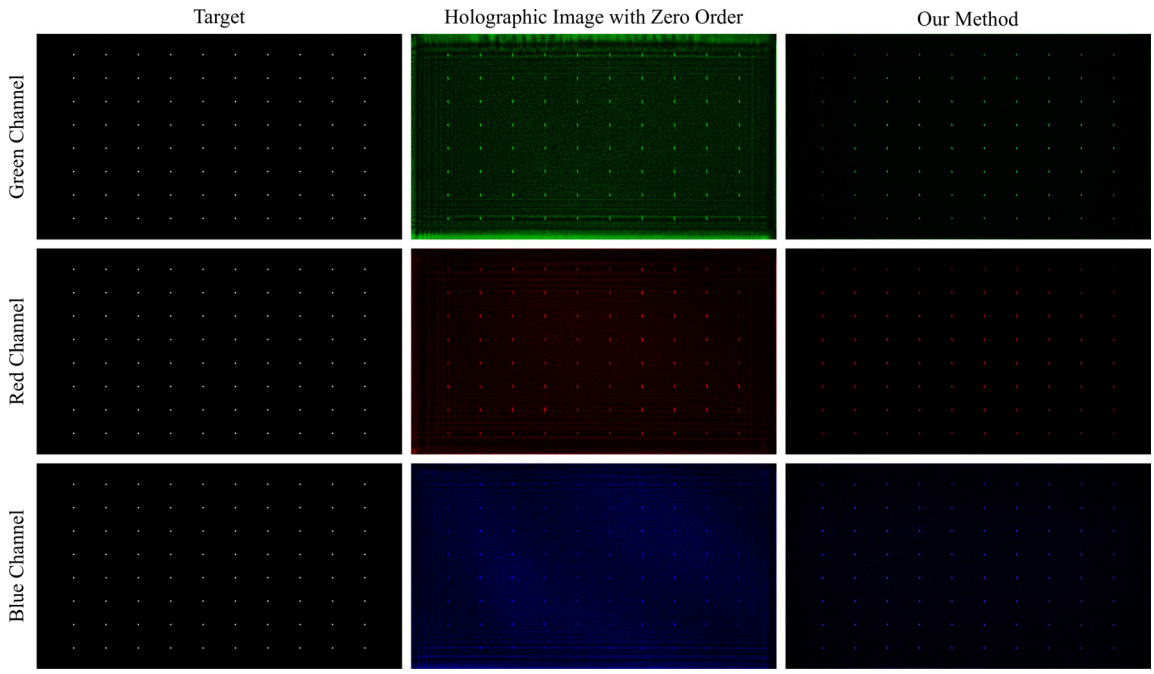


Fig. 1. We remove zero-order light in the proposed experimental setup. We show captures from our hardware prototype that validate doing so improves the image contrast substantially.

6 ADDITIONAL EXPERIMENTAL VALIDATION

6.1 Additional Experimental Captures

We found that the proposed method consistently outperforms existing phase retrieval methods across several test scenes. Wirtinger holography with DC order introduces undesirable ringing artifacts along the borders of the image. Wirtinger holography without the DC order mitigates this artifact but in doing so it produces severe granular noise across the image alongside global artifacts such as horizontal and vertical streaks which can be seen across images. Our proposed phase retrieval method effectively reduces the laser speckle to a finer resolution with a lighter intensity. Furthermore, it minimizes the impact of other forms of aberrations such as the aforementioned horizontal and vertical streaks. Figures 2 through 11 showcase holographic display captures from each of these phase retrieval methods and affirm these observations.

6.2 Additional 3D holographic display results

We demonstrate that the proposed hardware-in-the-loop optimization can be naturally extended to optimizing 3D holographic displays. We do this by applying the proposed method to a stack of 2D slices of a 3D volume. Note that the aberration approximator was not trained on these 2D slices but still manages to perform the desired

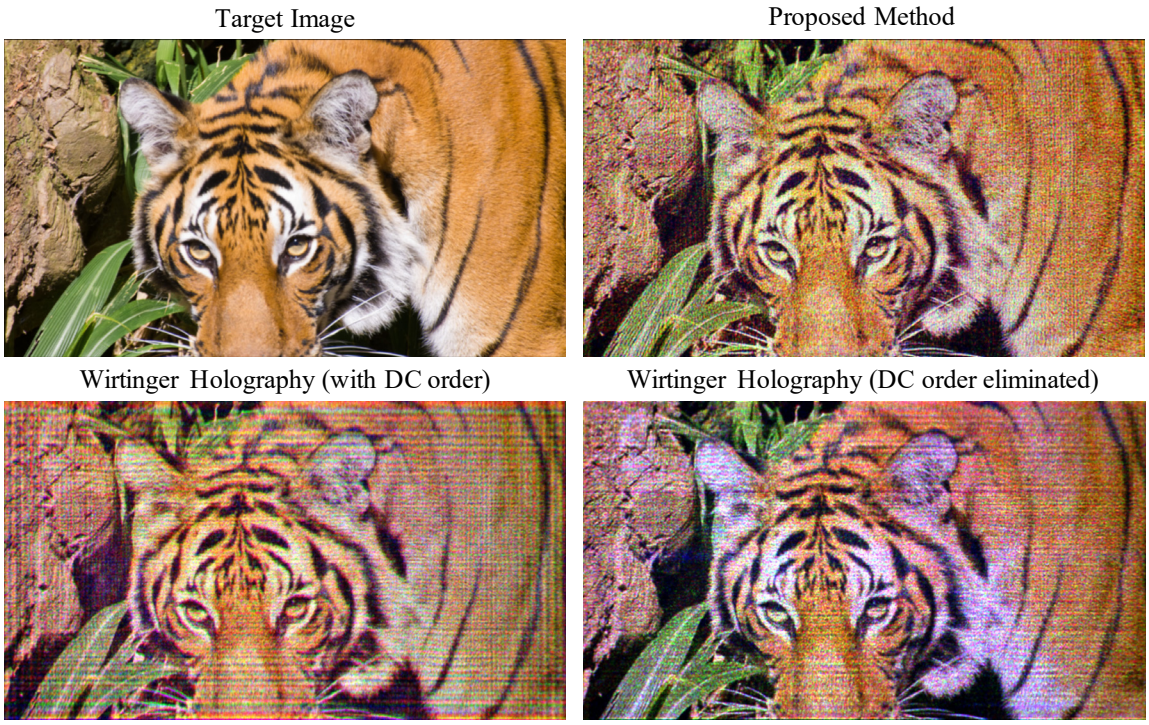


Fig. 2. Wirtinger holography with zero order included introduces an undesirable ringing artifact around the image. Removing the DC component reduces these ringing artifacts; however, we still find intense laser speckle throughout the image alongside global artifacts such as blue streaks that can be seen above the left eye of the tiger. The proposed method is able to significantly reduce the intensity of the laser speckle and decreases the impact of the previously mentioned streaks.

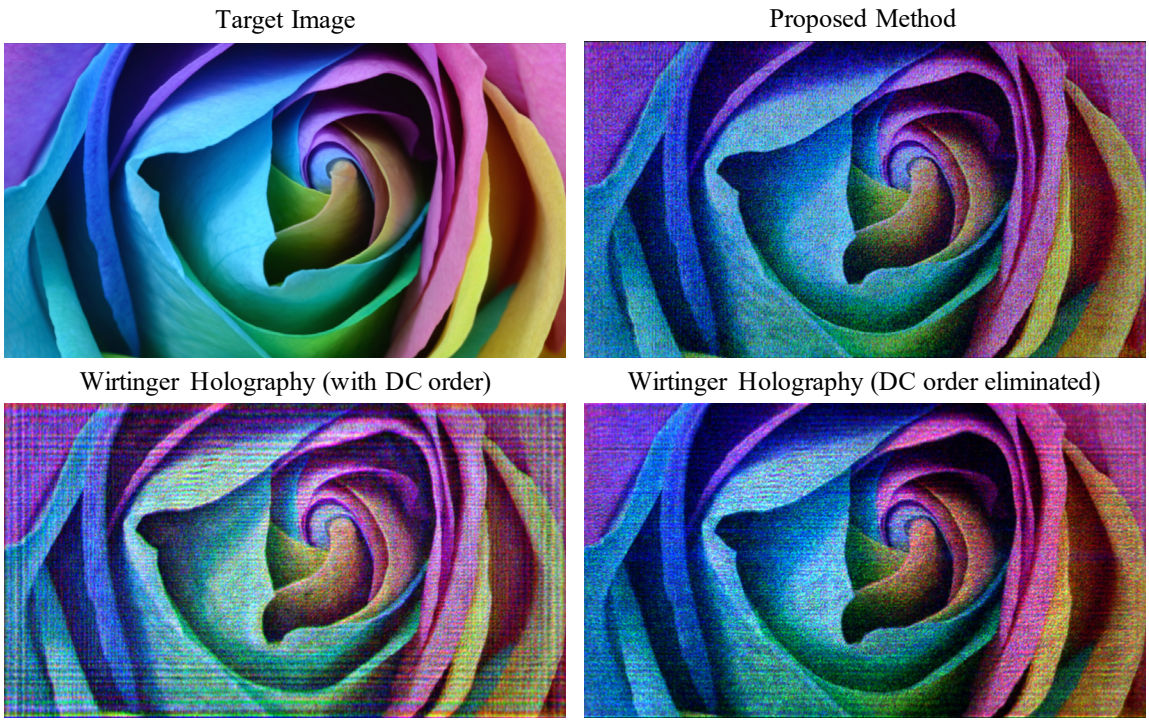


Fig. 3. If we observe the flat textured regions of the petals, we find that the proposed method significantly reduces the intensity of the coherent speckle compared to other methods. In Wirtinger holography with DC component eliminated, we find large granular noisy pixels. The proposed method significantly reduces the intensity and size of such pixels. Global artifacts such as horizontal streaks are also mitigated by the proposed method.

task. Qualitative results are shown in Figure 9 of the main paper and Figure 12. The proposed method reduces the holographic aberrations and allows for fine details to be seen at both near and far focus.

7 ONLINE-CAMERA PHASE OPTIMIZATION

We extend the proposed hardware-in-the-loop phase retrieval to an online-camera based optimization framework where new images acquired at each iteration are used for refining the aberration compensating phase holograms. Note that unlike the default hardware-in-the-loop framework, we are now separately refining an aberration approximator for each individual test image. Prototype results using this online optimization are shown in Figure 13. Although applying phase optimization with an active online-camera for each image frame produces improved holographic projections, our learned aberration approximator compensates for most aberrations after just a single refinement iteration and without having observed the test images as discussed in the main manuscript, thereby eliminating the need for additional refinement through online phase optimization.

8 ADDITIONAL ANALYSIS

8.1 Initialization and Termination criteria

Here we discuss the initialization and termination criteria for our hardware-in-the-loop phase retrieval framework.

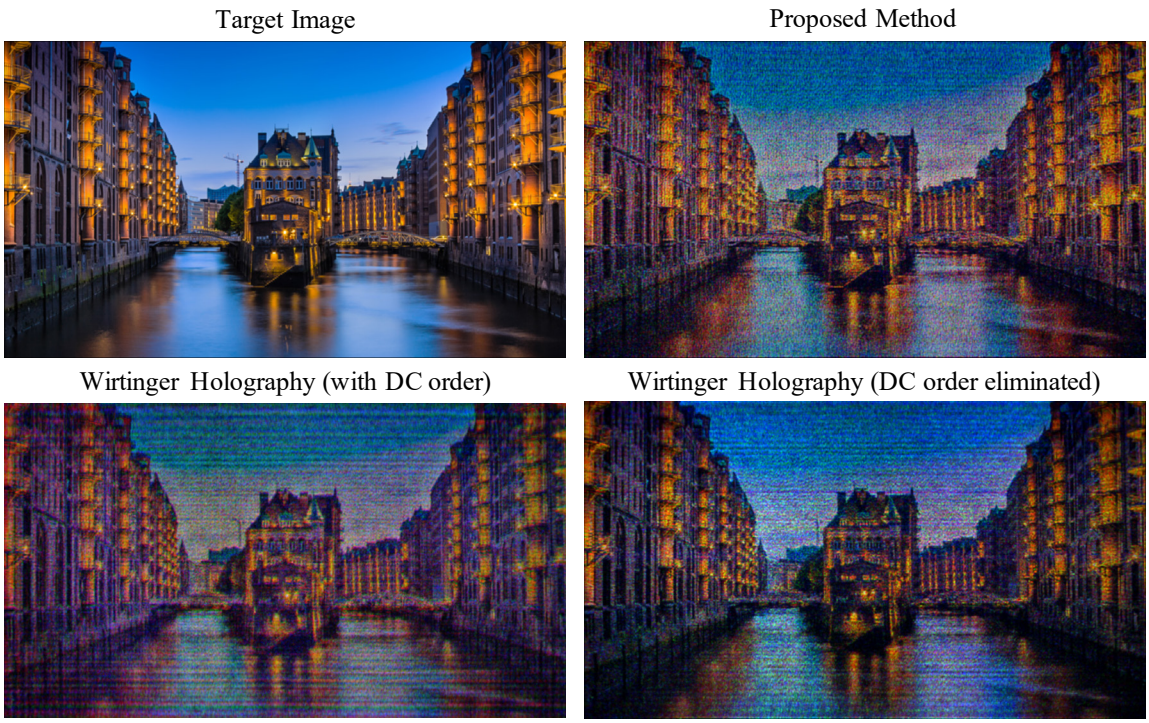


Fig. 4. The proposed method outperforms prevailing phase retrieval methods. If we observe the centre of the display capture using Wirtinger holography with DC order removed, we find several large noisy green pixels alongside visible dark lines across the scene. The proposed method mitigates these streaks and noisy green pixels. Laser speckle is reduced to a much finer resolution with a lighter intensity.

To solve the optimization as described in Algorithm 1 of the main manuscript, we start the optimization process with an initial guess of the phase value ϕ and the aberration approximation to be the identity function. We then alternate between the following two procedures for K iterations.

(1) We train the aberration approximator neural network to map phase holograms (assuming no aberrations) to holographic images as captured from the real hardware. Neural network training is done for 40000 steps.
(2) After training, we freeze the aberration approximator and use it to optimize for a new set of holograms that refine the errors produced by the phase holograms in the previous iteration. The optimization of aberration compensating phase holograms is terminated when the 1-LPIPS score surpasses 0.5 to ensure good perceptual quality of the final holographic display outputs.

We empirically find that running Algorithm 1 for $K = 1$ iterations already compensates for many aberrations.

9 ADDITIONAL DETAILS FOR ABERRATION APPROXIMATOR NETWORK

Our generator and discriminator architectures are shown in Figure 14. As described in the paper, our generator architecture is a variant of the U-Net [Ronneberger et al. 2015] with modifications that significantly improve its ability to learn the holographic aberrations. Our architecture design choices aim to allow the generator network to better learn the fine grained aberration details and color tones observed in the holographic captures.

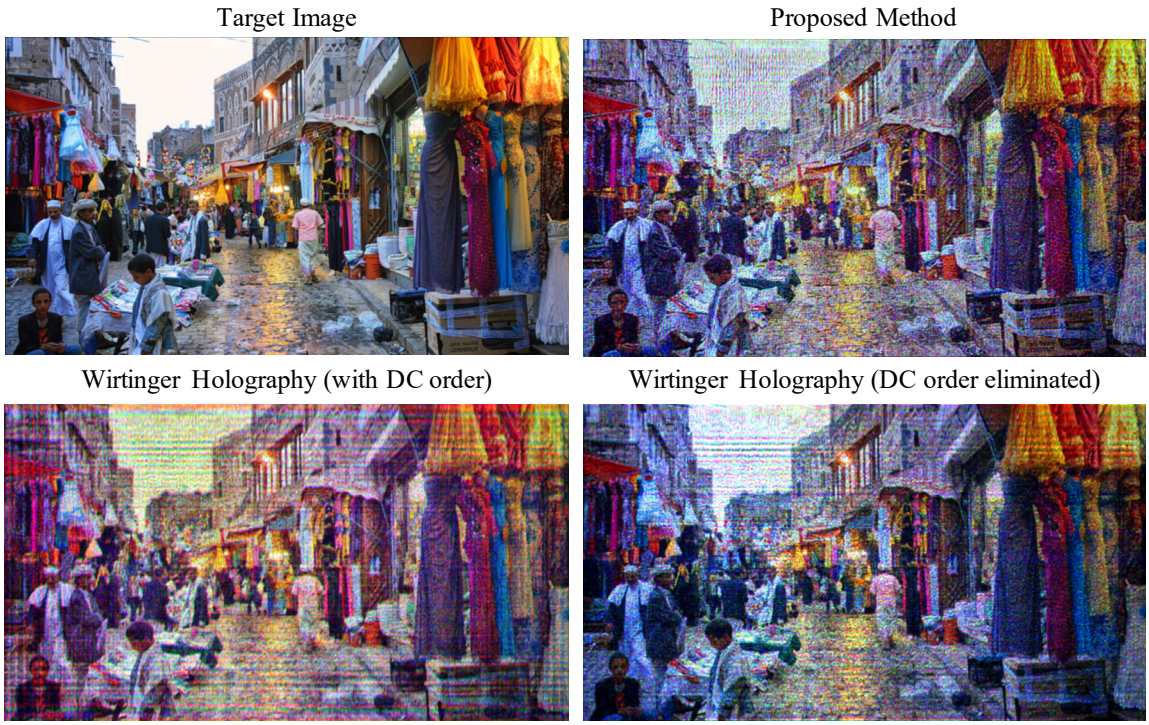


Fig. 5. The display capture for Wirtinger holography with DC order removed shows several dark lines across the image. The proposed phase retrieval method effectively removes observable instances of these lines.

Departing from the popular Pix2Pix generator architecture, we removed dropout which we found caused excessive regularization. We then added an additional convolution layer at the original 1080×1920 resolution with the aim of allowing finer details to be reproduced. Since we use single-batch training, we use instance normalization instead of batch normalization. We removed instance normalization from the first two encoding layers and last two decoding layers. Since instance normalization is to reduce the effects of image contrast [Ulyanov et al. 2016], we removed it from the higher resolution layers to allow the network to better learn color tones, but kept them at the lower resolution layers to facilitate network training. The discriminator architecture is a 94×94 PatchGAN. We deviate from traditional conditional GANs by conditioning our discriminator on the ideal simulated reconstruction instead of a semantic guide such as a segmentation map. For our perceptual loss we used the VGG-19 network with weights from the following source: https://www.kaggle.com/teksab/imagenetvggverydeep19mat.

We trained our network for 40000 iterations using the Adam optimizer and found training to be stable and that the network converged to a good point for aberration prediction. See Figure 16 for the training plots showing convergence of the proposed content loss, PSNR, SSIM, and 1 - LPIPS over the training period.

9.1 Baseline Comparisons

For our comparisons against alternative deep learning methods we used the following experimental setup. For Pix2Pix and Pix2PixHD we use their code with the default settings and train for 40000 iterations. For the U-Net comparisons we train for 40000 iterations using the Adam optimizer with learning rate 0.0002 and $\beta_1 = 0.5$ and

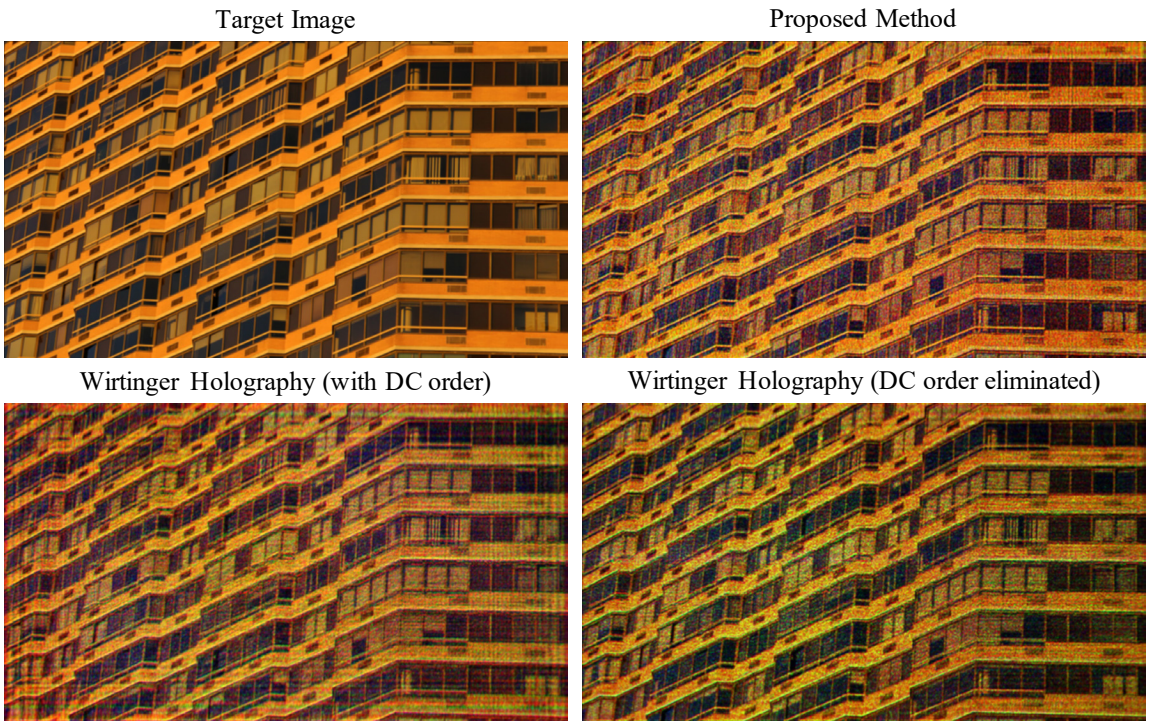


Fig. 6. Improved texture and reduced noise in the walls and windows of the building can be observed with the proposed method.

$\beta_2 = 0.999$, and we use the same architecture as described in Ronneberger et al. [2015] except with half the number of filters at each layer due to memory constraints. For Pix2PixHD we use the default settings for their network architectures and loss functions except that we reduce the number of filters in the first layer of the generator and discriminator from 64 to 32 due to memory constraints.

In addition to Figure 6 from the paper, Figures 19 and 20 show additional comparison examples. We again see that the U-Net methods are unable to capture the aberration details. Although Pix2Pix and Pix2PixHD produce believable aberrations, the error maps show a large mismatch between their network predictions and the actual target display.

Overall, we found that using these methods out-of-the-box did not work for our task of predicting fine aberration details in the holographic captures. As such, we proposed the network and loss function as described in the main paper towards the desired task. The ablation study in the main paper and in the following section further validates our design choices.

9.2 Ablation Study

We demonstrate the importance of our network architecture and loss function through our ablation study, see Table 1 for quantitative results. We found that careful adjustments were necessary when designing the aberration approximator architecture and loss function. Although the proposed network is capable of outputting high fidelity aberration predictions, we found that a good loss function design was necessary to avoid bad local minima.

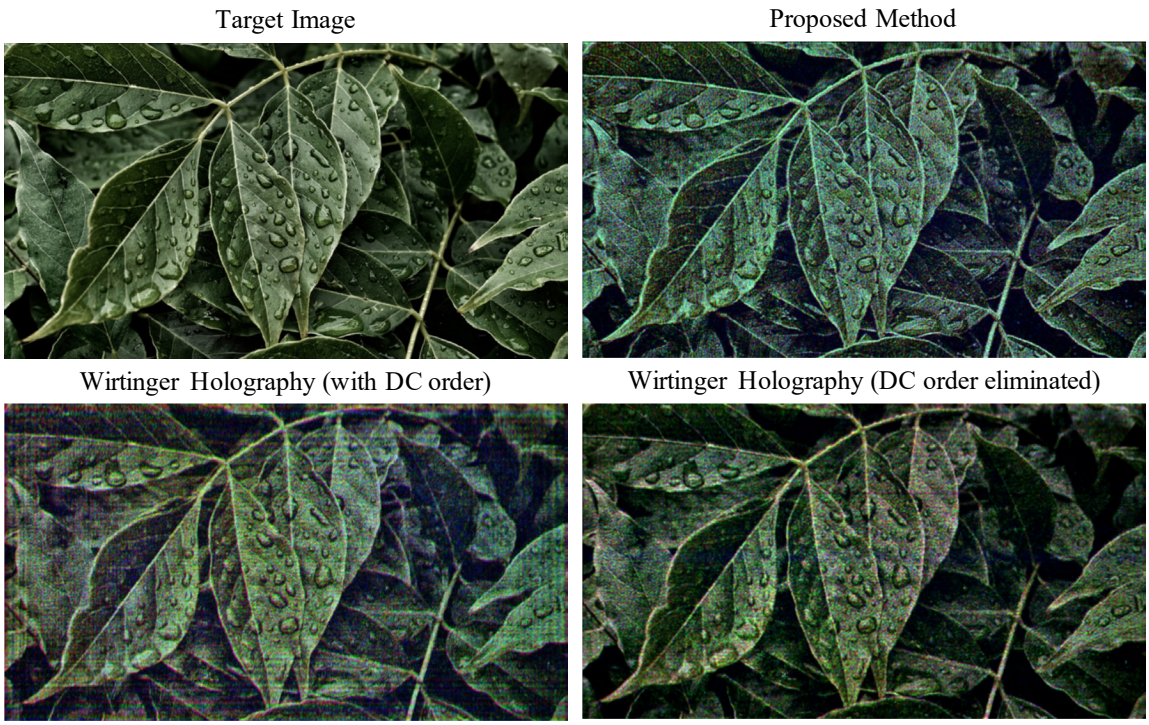


Fig. 7. The proposed method mitigates the streak artifacts and reduce the impact of aberrations seen in the Wirtinger Holography methods.

Table 1. Quantitative results for ablation on loss function and architecture modifications. The proposed design achieves the best perceptual quality. Although removing the adversarial and perceptual losses increases PSNR and SSIM, it causes a decrease in perceptual quality as can be seen in the 1-LPIPS score.

	PSNR (dB)	SSIM	1-LPIPS
Proposed	29.6	0.831	0.953
Proposed (unconditional)	24.5	0.628	0.825
Proposed ($\lambda_{\ell_1} = 0$)	23.1	0.574	0.808
Proposed ($\lambda_{\text{Perc}} = 0$)	30.2	0.841	0.929
Proposed ($\lambda_{\text{Adv}} = 0$)	30.5	0.850	0.920

Specifically, we found that using an unconditional GAN or setting $\lambda_{\ell_1} = 0$ caused the network to converge on a minima where the predicted aberration pattern is almost the same across images, see Figure 18 for qualitative examples. See also Figure 17a and Figure 17b for examples of the mismatch between the predicted aberration and the actual holographic display.

We found that using a combination of ℓ_1 loss and a conditional GAN avoided this local minima. When only using ℓ_1 and perceptual loss we found that the predicted aberration patterns had a slight blur to them, see Fig 17d. On the other hand, using only ℓ_1 and adversarial loss produced aberration predictions that look accurate at first glance, but upon closer observation the laser speckle patterns are often mismatched with the target, see Fig 17d. Note that, although the PSNR increases in Table. 1 when $\lambda_{\text{Perc}} = 0$ or $\lambda_{\text{Adv}} = 0$, these observable inaccuracies

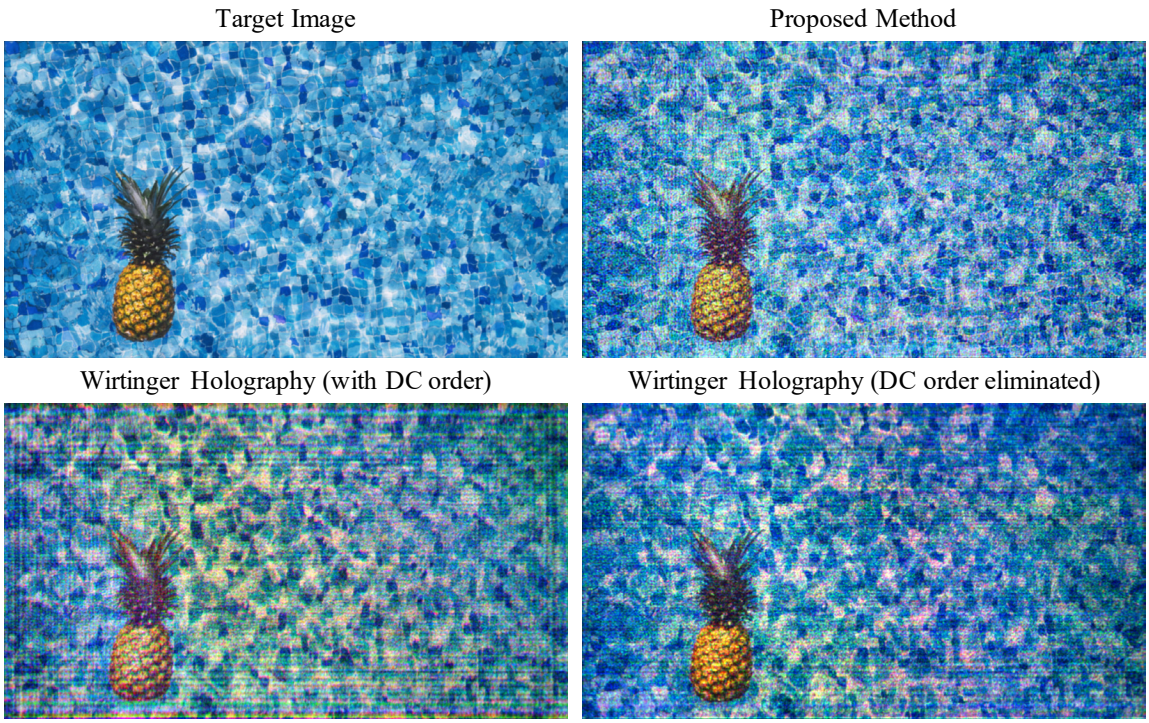


Fig. 8. Although uniform background textures like this swimming pool floor reveal some streak artifacts remaining in the proposed method, the impact of the aberration is still greatly reduced compared to the other two methods.

indeed correspond with a lower 1-LPIPS score. Since LPIPS has been shown by Zhang et al. [2018] to be a better measure of perceivable quality than PSNR or SSIM, the proposed design's top LPIPS performance in addition to good PSNR and SSIM performance demonstrates its superiority over the alternatives. As such, the proposed combination of ℓ_1, perceptual, and adversarial loss using a conditional GAN regime provides the best aberration predictions that generalize to unseen images, and this is evidenced by the accuracy of the proposed method when used for hardware-in-the-loop training (Table 2 and Table 3 in the main document).

9.3 Simulated Phase Retrieval with Aberration Approximator

Figure 21 shows additional simulation results that demonstrate that the aberration approximator allows for effective phase hologram optimization within the Wirtinger framework.

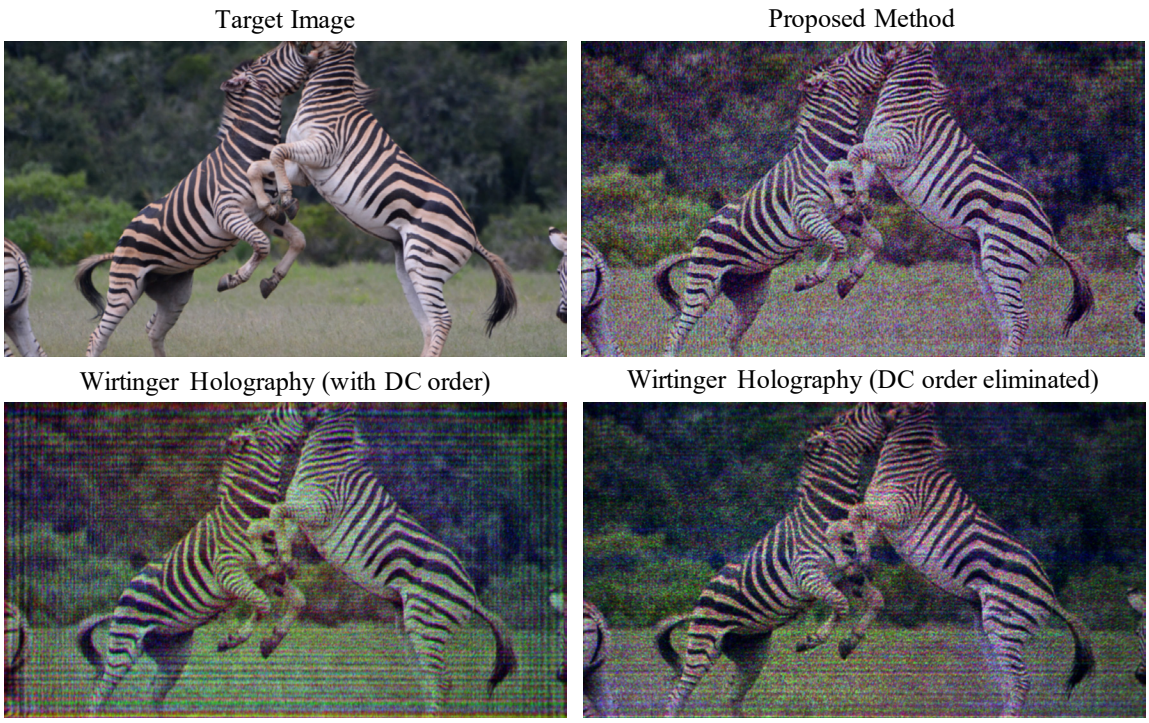


Fig. 9. The streaks in the grassy field and background trees are greatly reduced with hardware-in-the-loop optimization. Aberrations found on the zebras are also reduced.

REFERENCES

Praneeth Chakravarthula, Yifan Peng, Joel Kollin, Henry Fuchs, and Felix Heide. 2019. Wirtinger holography for near-eye displays. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 213.

Praneeth Chakravarthula, Yifan Peng, Joel Kollin, Felix Heide, and Henry Fuchs. 2020. Computing high quality phase-only holograms for holographic displays. In *Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR)*, Vol. 11310. International Society for Optics and Photonics, 1131006.

Joseph W Goodman. 2005. *Introduction to Fourier optics*. Roberts and Company Publishers.

Kyoji Matsushima and Tomoyoshi Shimobaba. 2009. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Optics express* 17, 22 (2009), 19662–19673.

Reinhold Remmert. 2012. *Theory of complex functions*. Vol. 122. Springer Science & Business Media.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

Fabin Shen and Anbo Wang. 2006. Fast-Fourier-transform based numerical integration method for the Rayleigh-Sommerfeld diffraction formula. *Applied optics* 45, 6 (2006), 1102–1110.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *ArXiv* abs/1607.08022 (2016).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint* (2018).

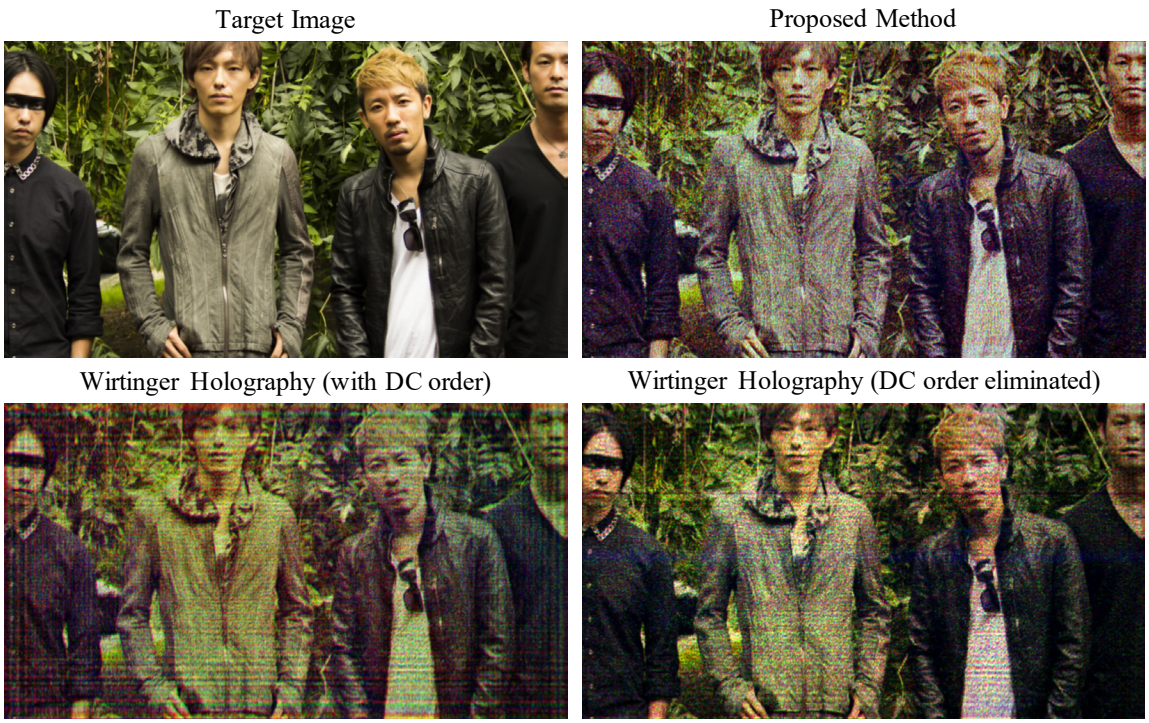


Fig. 10. The proposed method reduces the streak aberrations which allows for better visualization of the human faces and hair textures seen in this example.

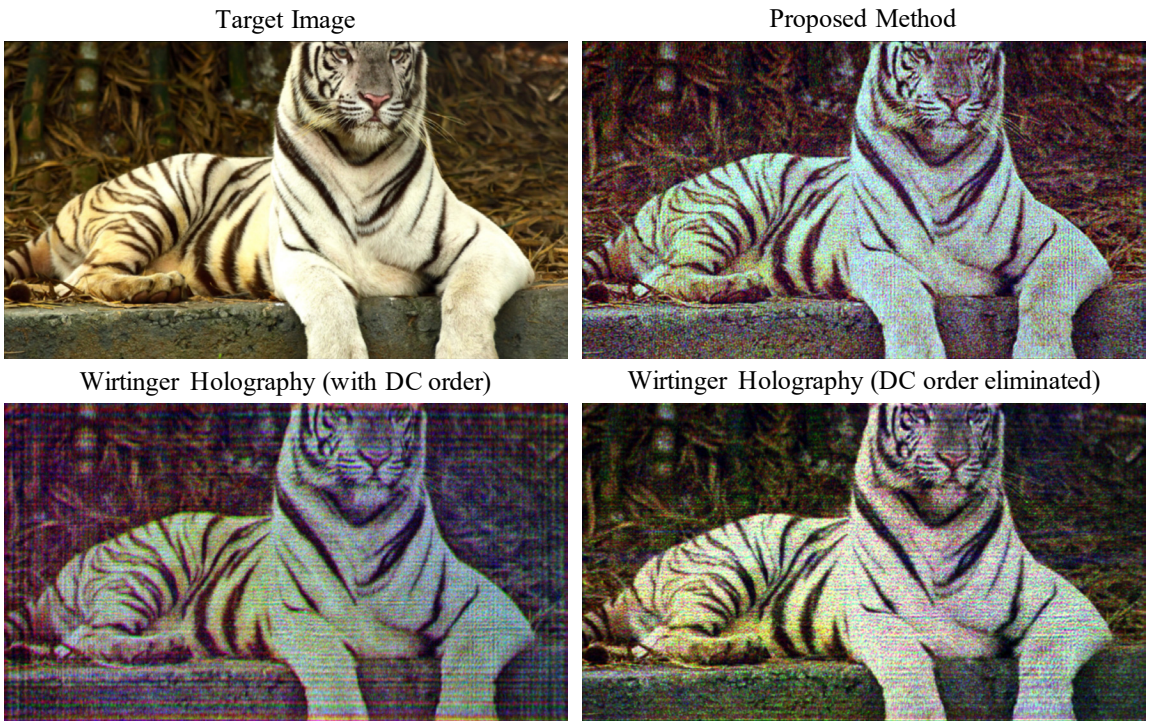


Fig. 11. The details in the face of the tiger can be better seen after optimization with the proposed method.

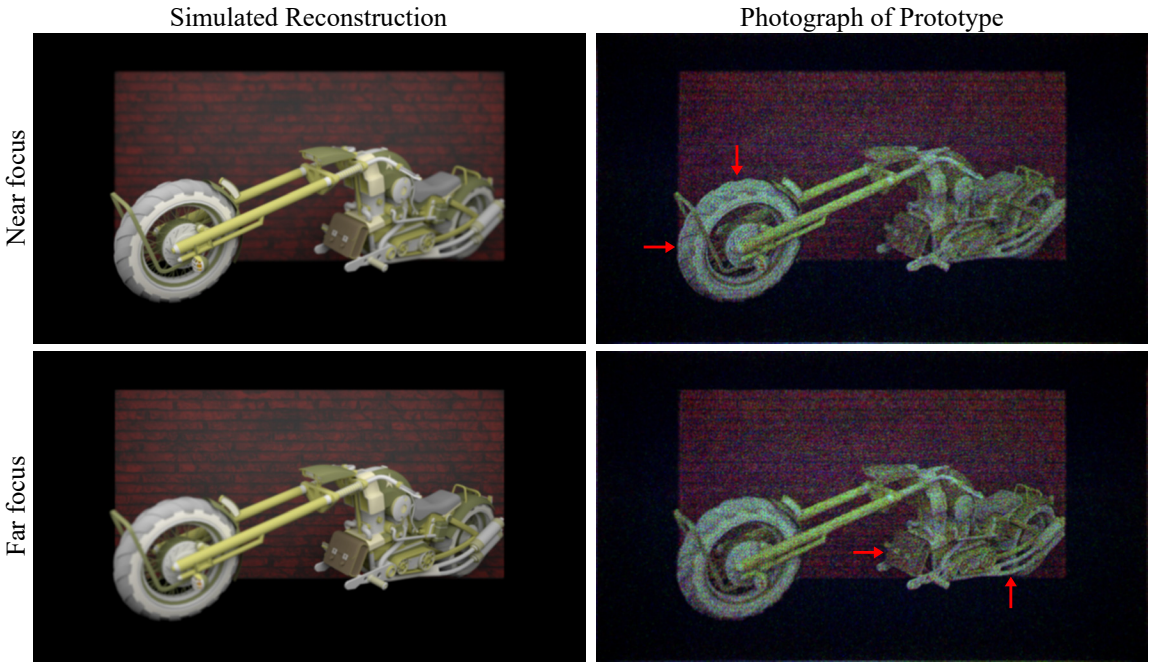


Fig. 12. After hardware-in-the-loop optimization we can see enhanced color and sharpness detail in the captured 3D holographic display. For example the grooves on the front wheel can be seen in the near focus case, and the bag and exhaust pipes are sharp in the far focus case.

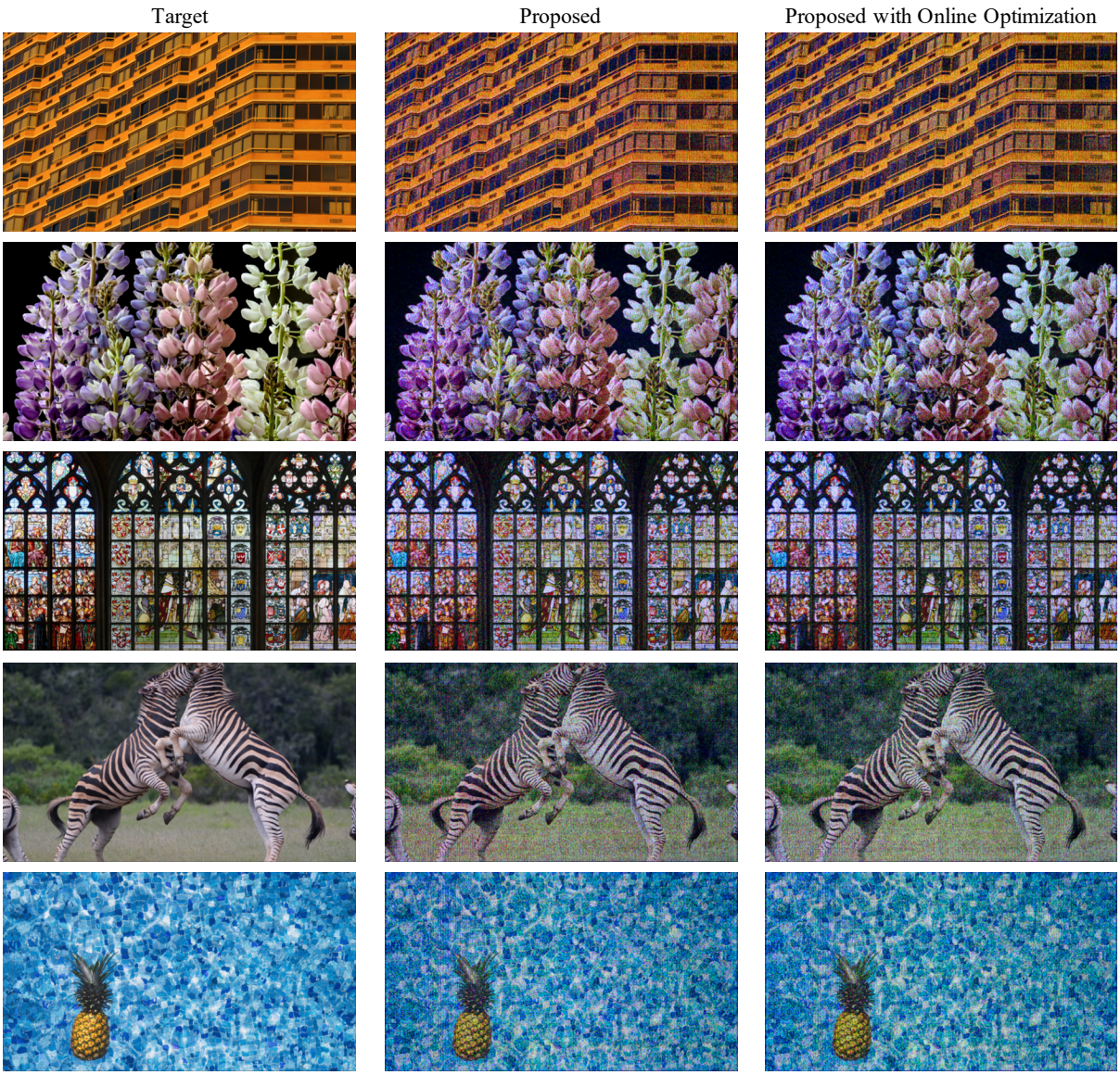


Fig. 13. We demonstrate the our method can be extended towards online phase optimization by refining the aberration approximator on images seen during testing. The middle column displays holographic captures after our hardware-in-the-loop optimization as described in the main paper. The right column displays holographic captures after refining on each individual image through online optimization. Examples of improvements can be observed the finer curtain details for the first row and the sharper flower details in the second row. The results demonstrate the viability of this framework for online phase retrieval.

Fig. 14. Generator and discriminator network architecture. In the table, "conv-k(a)-s(b)-IN-LRelu" represents a convolution layer with an $a \times a$ kernel window, using stride b, followed by instance normalization and a Leaky Relu ($\alpha = 0.02$) activation function. We use convT to denote transposed convolution. Our generator architecture is based on the U-Net architecture with substantial modifications. Our discriminator is conditioned on the same input into the generator, the ideal simulated reconstruction.

Generator

Layer Name	Type	Channels
Input	Ideal Simulation	3
down1	conv-k5-s2-LRelu	64
down2	conv-k5-s2-LRelu	128
down3	conv-k5-s2-IN-LRelu	256
down4	conv-k5-s2-IN-LRelu	512
down5	conv-k5-s2-IN-LRelu	512
down6	conv-k5-s2-IN-LRelu	512
down7	conv-k5-s2-IN-LRelu	512
down8	conv-k5-s2-IN-LRelu	512
up1	convT-k4-s2-IN-LRelu	512
concat1	concat(down7,up1)	1024
up2	convT-k4-s2-IN-LRelu	512
concat2	concat(down6,up2)	1024
up3	convT-k4-s2-IN-LRelu	512
concat3	concat(down5,up3)	1024
up4	convT-k4-s2-IN-LRelu	512
concat4	concat(down4,up4)	1024
up5	convT-k4-s2-IN-LRelu	256
concat5	concat(down3,up5)	512
up6	convT-k4-s2-IN-LRelu	128
concat6	concat(down2,up6)	256
up7	convT-k4-s2-LRelu	64
concat7	concat(down1,up7)	128
up8	convT-k4-s2	32
Output	conv-k4-s1	3

Discriminator

Layer Name	Type	Channels
Input	Ideal Simulation and Generator Output	6
down1	conv-k4-s2-LRelu	64
down2	conv-k4-s2-IN-LRelu	128
down3	conv-k4-s2-IN-LRelu	256
down4	conv-k4-s2-IN-LRelu	512
down5	conv-k4-s2-IN-LRelu	512
Output	conv-k4-s2-LRelu	1

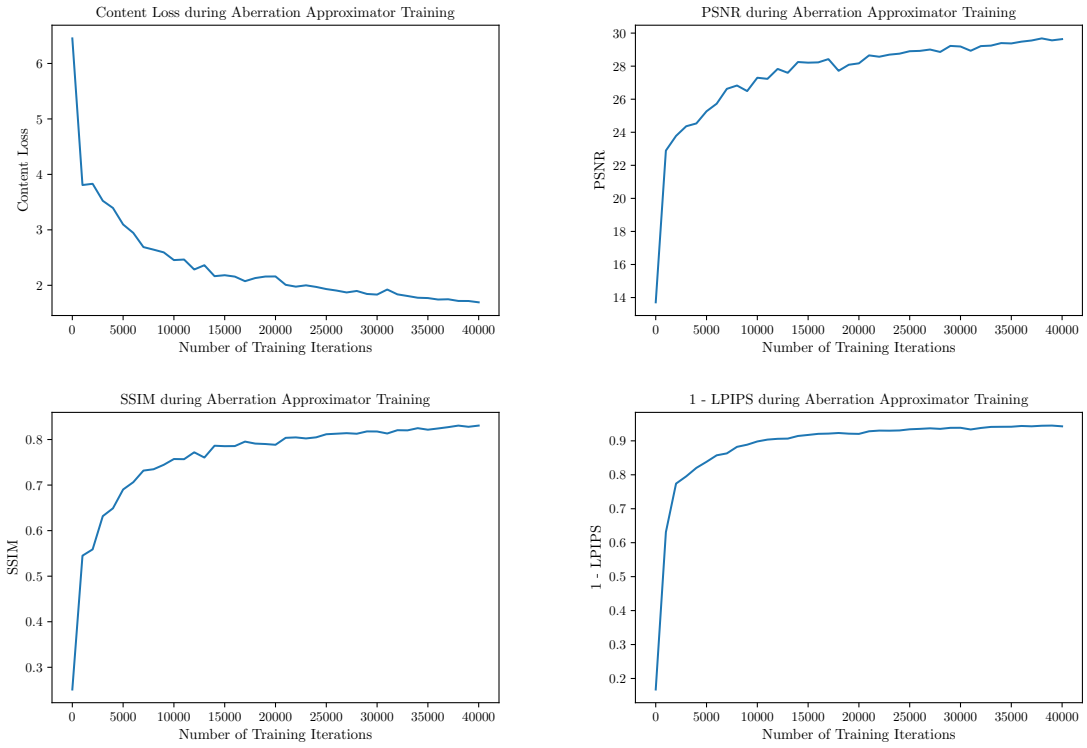
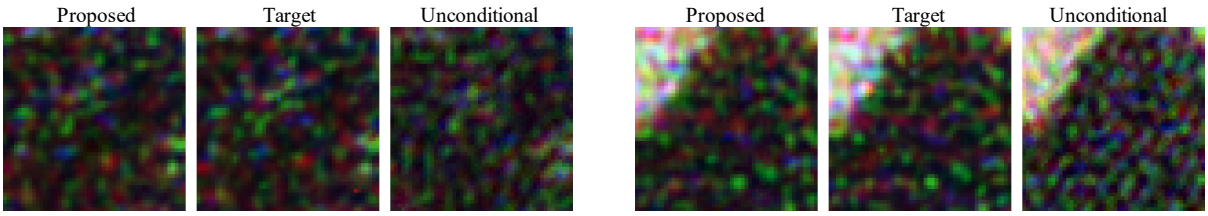
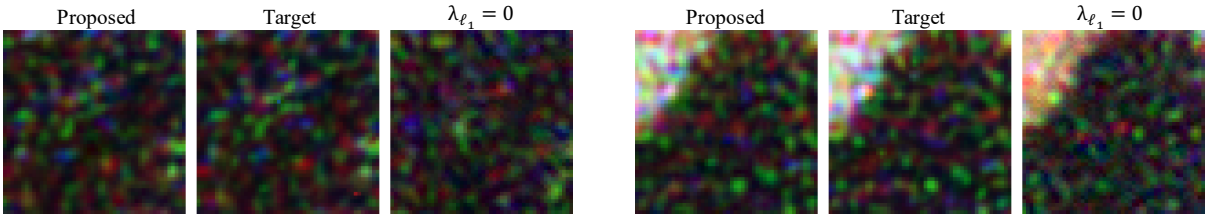


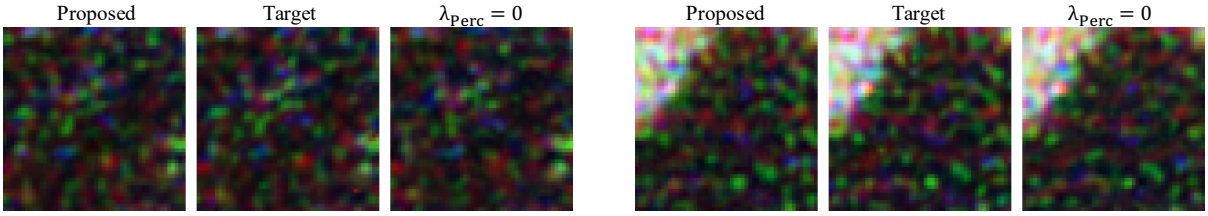
Fig. 16. Training plots when training the aberration approximator with the proposed loss function. Our network converges upon a good solution for prediction holographic aberrations with high performance across all image quality metrics.



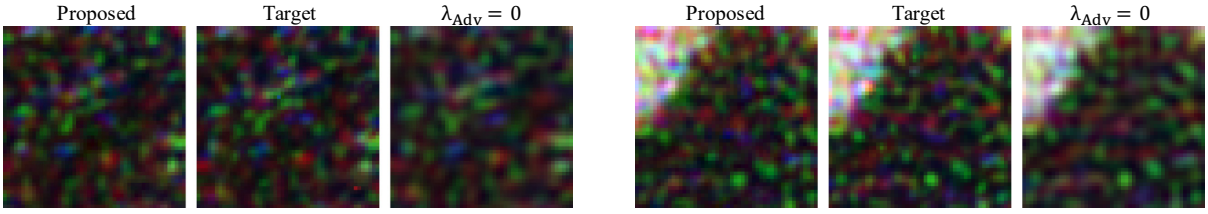
(a) Aberration prediction patches for proposed method using an unconditional discriminator. Without being conditioned on the ideal simulated reconstruction the network training is more prone to falling into a local minima where the predicted aberrations do not match the actual aberrations.



(b) Aberration prediction patches for proposed method without ℓ_1 loss. Without the ℓ_1 loss the network is also prone to falling into local minima that does not accurately predict the holographic aberrations.



(c) Aberration prediction patches for proposed method without perceptual loss. Without perceptual loss the network predictions are often misaligned with the target holographic aberrations. In these examples, the green laser speckles are either missing or are predicted in the wrong location.



(d) Aberration prediction patches for proposed method without adversarial loss. Without the adversarial loss the network the predicted aberrations are all blurrier than actual holographic aberrations.

Fig. 17. Qualitative comparison of ablated methods against the proposed method for select aberration patches.

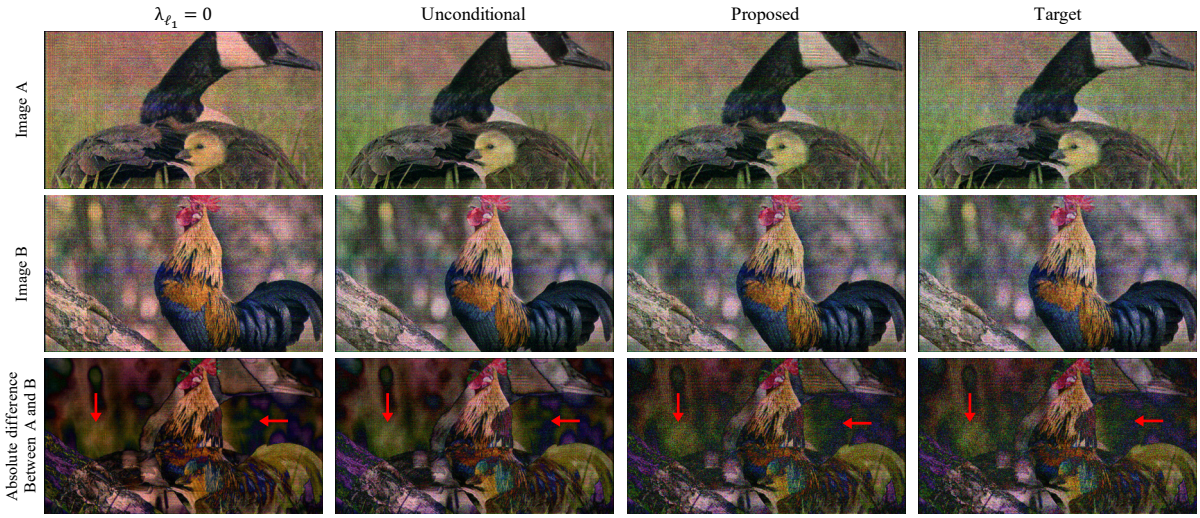


Fig. 18. Without the support of the ℓ_1 loss component or the conditional input into the discriminator, we observed the network training to be more unstable and prone to encountering local minima. The left two columns shows the network making similar aberration predictions regardless of the input image and failure to learn the fine aberration differences between images. Observe that by taking the absolute difference between two different image scenes, the aberrated noise pattern is almost completely removed, indicating that the predictions are almost identical regardless of the input scene. In contrast, taking the same difference between the targets shows many differences even in the aberration noise patterns. Our proposed method avoids this local minima as shown in the second column from the right, as both the predicted aberrations and the absolute difference map match the target's. Please zoom in to see details.

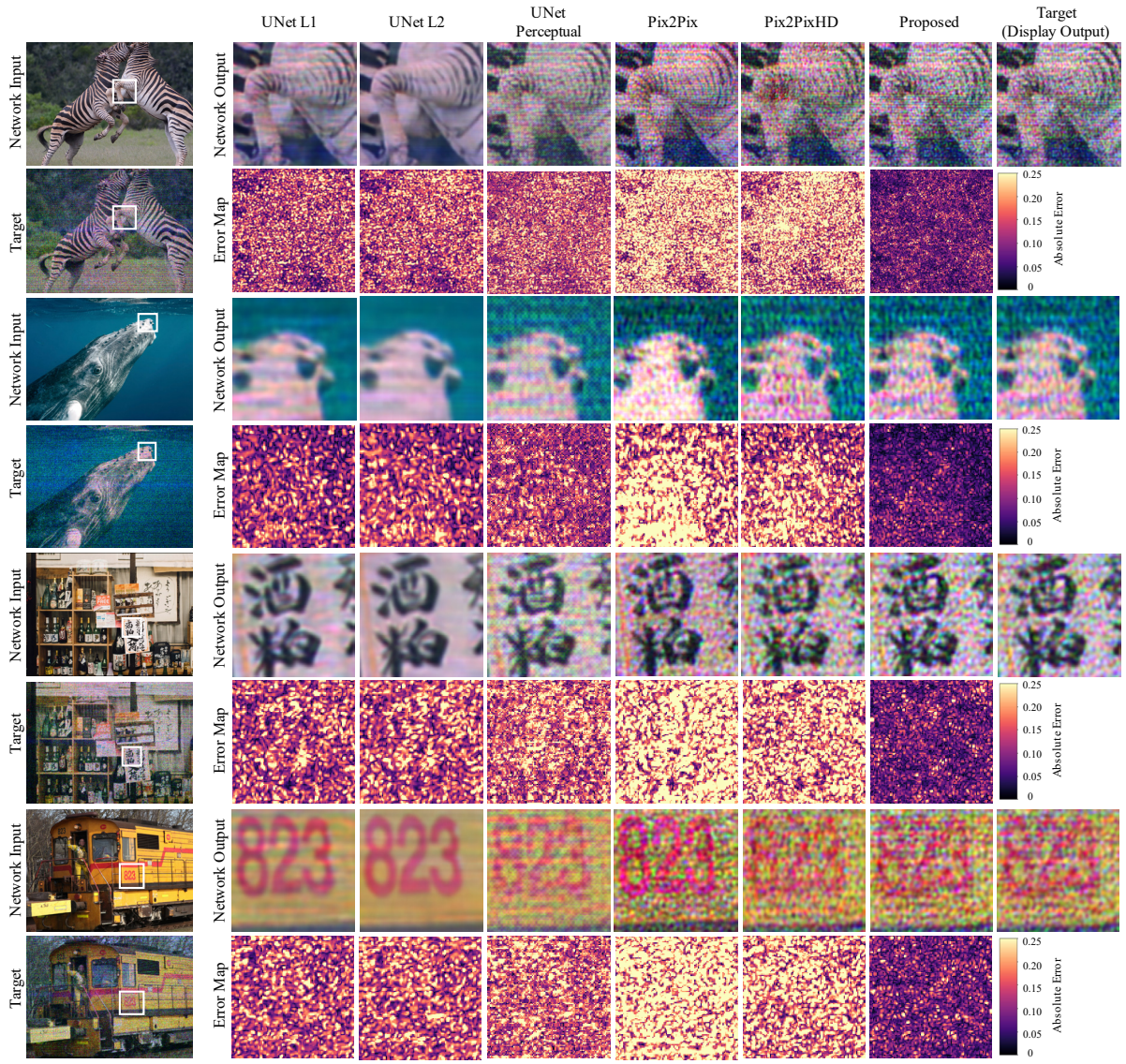


Fig. 19. Comparison of different baseline methods and proposed method for the Aberration Approximator. The U-Net approaches fail to learn the diverse aberration patterns. Pix2Pix is better but makes many mistakes as can be seen in the error maps. Pix2PixHD is similar but also hallucinates additional artifacts, an example can be seen in the first row.

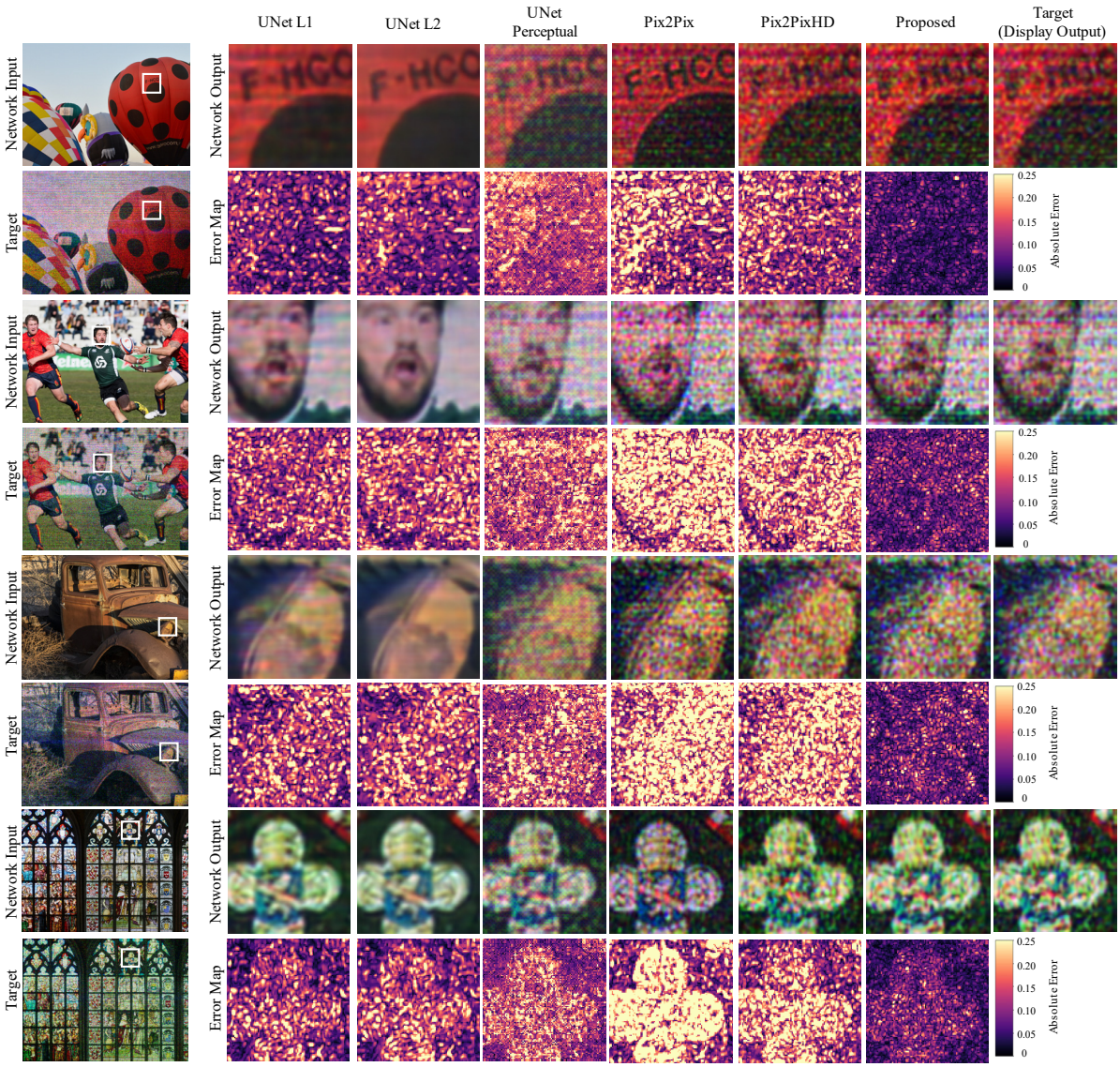


Fig. 20. Comparison of different baseline methods and proposed method for the Aberration Approximator. The U-Net approaches fail to learn the diverse aberration patterns. Examples of mistakes made by Pix2Pix can be seen in the first row where the letters are not sufficiently aberrated and the fourth row where the color tones do not accurately match the display. The error maps for Pix2PixHD reveal many aberration prediction mismatches despite the believable predictions at first glance.

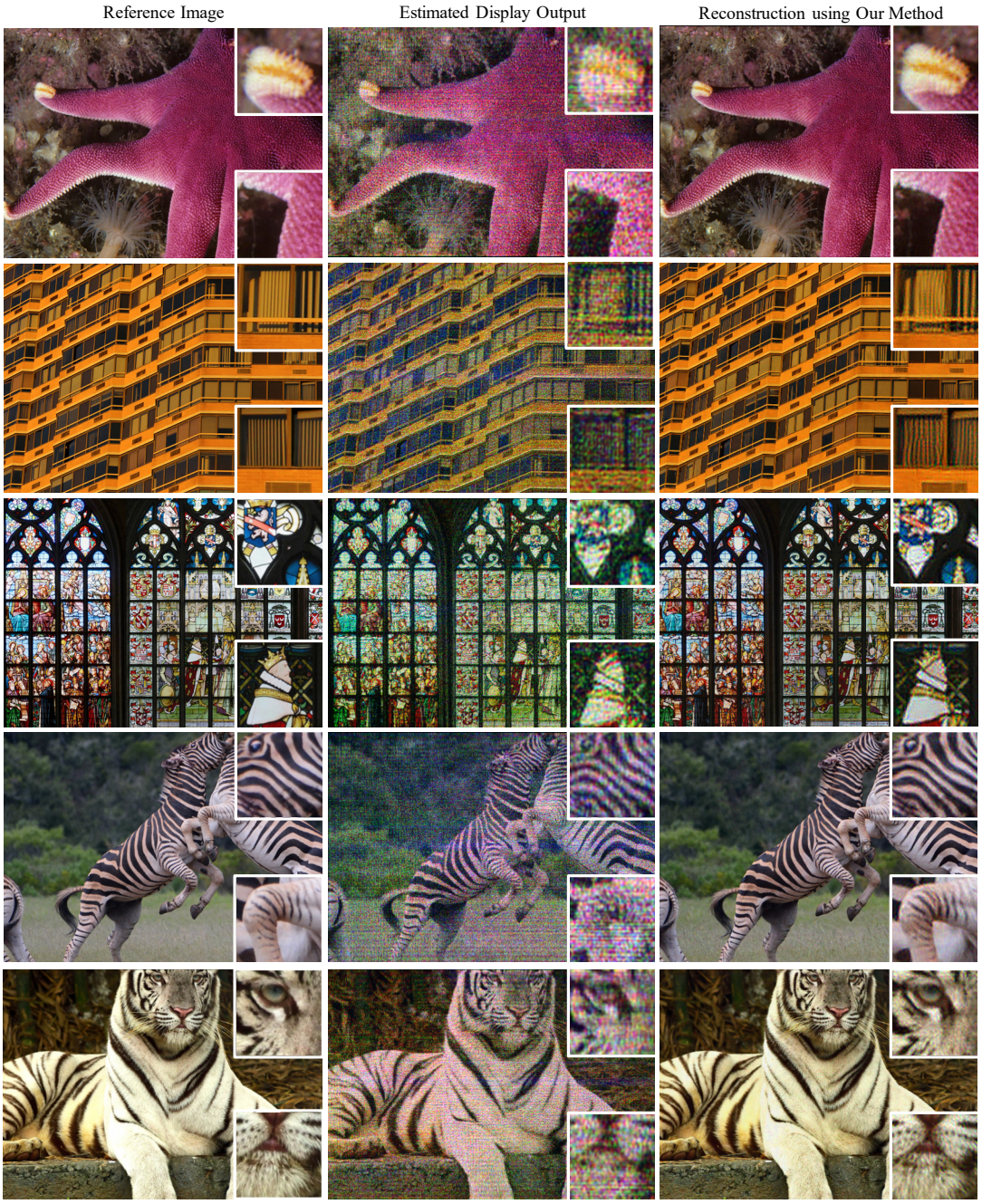


Fig. 21. From left to right, we show the reference image, estimated display output as predicted by the aberration approximator network and the reconstruction of the optimized aberration-compensating hologram as generated by the proposed method. This figure provides additional results to Figure 7 in the main manuscript.