Configurable Holography: Towards Display and Scene Adaptation

YICHENG ZHAN[1], LIANG SHI[2], WOJCIECH MATUSIK[2], QI SUN[3], AND KAAN AKŞIT[1,*]

[1] University College London, Gower Street, London, UK, WC1E 6BT
[2] Massachusetts Institute of Technology, MA 02139, Massachusetts, USA
[3] New York University, New York, USA
[*] kaanaksit@kaanaksit.com

Compiled June 21, 2026

Rendering holograms for holographic displays is often an iterative and computationally costly process. Emerging learned holography methods have alleviated this bottleneck by enabling fast hologram rendering with improved reconstruction quality. However, existing methods still depend on fixed display hardware and scene parameters, requiring retraining for each new configuration. This limits rapid adaptation to different visual needs, including scene brightness, user focus preference, and hardware compatibility. We introduce *Configurable Holography*, a learned CGH framework in which a single model adapts to diverse display-scene parameters through explicit conditioning, eliminating the need for retraining. As a prototype, we present a configurable structure and derive a family of models that continuously adapt to propagation distance, volume depth, peak brightness, pixel pitch, and wavelength. To further improve efficiency, we incorporate auxiliary monocular depth estimation for depth-aware 3D hologram synthesis from RGB-only inputs and apply knowledge distillation for interactive inference. Our extensive simulation and hardware experiments on three holographic display prototypes with different combinations of configurations show on-par reconstruction quality with existing methods, offering up to 2× speed-up in fp32. Our work represents an initial step toward flexible, general-purpose learned holography systems that can seamlessly adapt across diverse hardware and user-specific visual requirements.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Holographic displays [3] support optical focus cues and multiple perspectives, promising authentic immersive Three-Dimensional (3D) visual experiences as a potential future display technology. However, rendering visuals for holographic displays remains computationally demanding and slow [4]. Emerging learned holography methods [1] promise to accelerate and improve visual quality in the holographic displays. However, existing learned holography models are inflexible as they require training a dedicated model for each set of display-scene parameters. This constraint becomes a bottleneck in practice. For instance, users demand instant, continuous control over focus range and brightness levels; those users with prescriptions also require focus adjustments for clearer images; and developers building displays need full tunability at interactive-rate across system parameters to accelerate the development processes of display prototypes. *Meeting these diverse requirements with separate models would cause substantial overhead in model management, deployment speed, initialization time, and training, compromising seamless visual experiences. This makes model configurability an important capability for future holographic displays.* We also acknowledge that for a deployed holographic display with fixed focal length, per-configuration models remain practical.

We advocate *Configurability* as an important objective for learned Computer-Generated Holography (CGH). We define configurability as the ability of a single learned model to *adapt* its hologram synthesis behavior according to the requested display-scene parameters at inference time, without retraining for each configuration. In this view, learned CGH is not merely a fast alternative to optimization under one setting, but a programmable hologram generator that can accommodate parameter changes that naturally arise across users, scenes, and devices. Achieving this goal is non-trivial because display-scene parameters influence diffraction physics and perceived 3D appearance; a configurable model must therefore balance generality, reconstruction fidelity, and efficiency.

Building on this concept, our work depicted in Fig. 1 focuses on interactive 3D hologram synthesis across a continuous range of various display-scene parameters using a single learned model. We introduce a highly configurable network structure

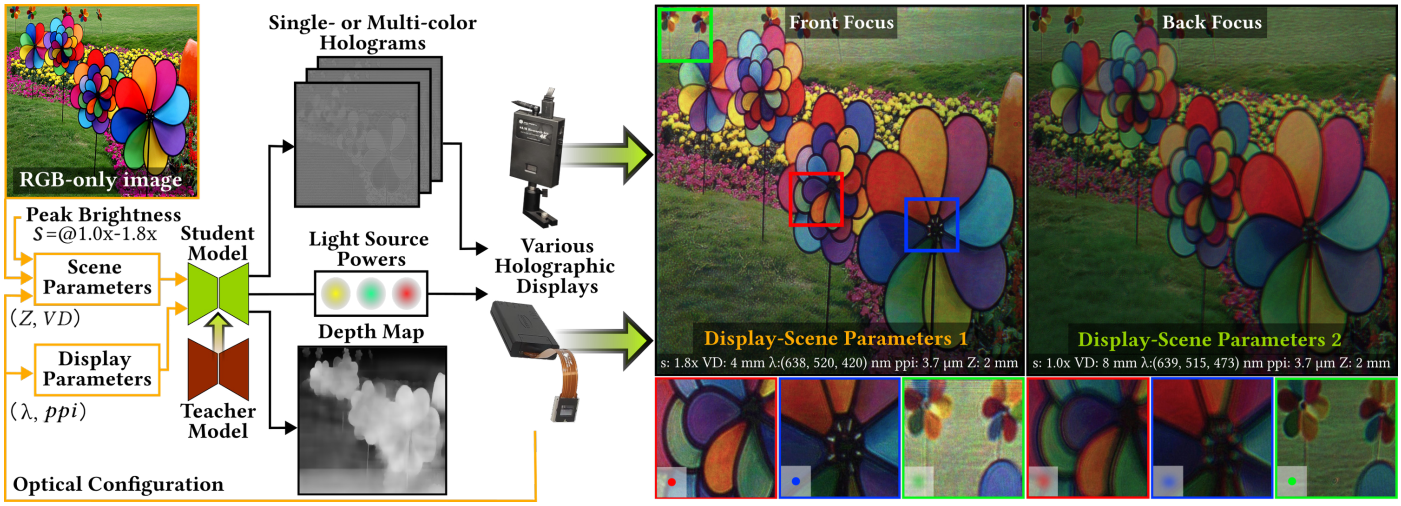


Fig. 1. Our configurable holography model supports a range of display–scene parameters without retraining, including peak brightness (s), propagation distance (Z), volume depth (VD), working wavelength (λ), and pixel pitch (ppi). We distill our model into a student model, achieving 2x speed up w.r.t. literature [1]. Our model can synthesize single or multi-color 3D holograms from RGB-only 2D input images by internally predicting depth, simply eliminating the need for depth input (Source Image: [2]).

that synthesizes conventional single-color [5] and multi-color [6–8] holograms while being conditioned on scene parameters (propagation distance, volume depth, peak brightness), and display parameters (pixel pitch and wavelength). Alongside configurability, we also explore two strategies to maximize the efficiency of learned CGH methods. First, we treat Monocular Depth Estimation (MDE) as an auxiliary training task to improve the 3D accuracy of hologram synthesis. Importantly, our goal is not to compete with advanced MDE methods, but to demonstrate that depth information can serve as an effective additional supervision signal for accurate 3D hologram synthesis from RGB-only inputs with no depth, the most common form of media. Second, we apply Knowledge Distillation (KD) [9] to distill a compact and configurable student model that is substantially accelerated for interactive inference and preserves reconstruction quality. While our models demonstrate that configurable CGH is feasible, we emphasize that this work is a prototype and configurability itself warrants more rigorous and scaled evaluation. Accordingly, beyond reporting performance, we propose several Research Question (RQ): why is achieving configurability important and non-trivial? Why are MDE and KD beneficial to CGH? What limitations does our method exhibit, and how might future work address them? We hope these observations will help readers reason about configurable CGH and inspire future research. Our contributions are as follows:

- **Configurable Holography.** We introduce configurability as a target capability for the learned CGH method and present a configurable model structure as a prototype. From which we derive a family of models supporting both single- and multi-color 3D holograms across a range of display-scene parameters, including propagation distance, volume depth, peak brightness, wavelength, and pixel pitch.
- **Advancing efficient and accurate 3D Hologram Synthesis from RGB-only inputs.** We unearth the correlation between depth estimation and 3D hologram synthesis tasks in learned methods. Our model leverages this correlation and adopts multitask learning with hard-parameter sharing [10] to convert RGB-only Two-Dimensional (2D) images to accurate 3D holograms. We also apply KD to train a compact student model that achieves up to 2× faster hologram synthesis than prior learned approaches [1] under 32-bit precision (fp32) while preserving reconstruction quality.
- **Evaluation and empirical validation.** We conduct extensive quantitative and qualitative experiments comparing against existing learned CGH methods and validate our findings on three physical holographic display prototypes.

Our design targets conventional holographic displays and therefore inherits their typical FoV and eyebox constraints [11]; extending configurability to emerging display architectures [12–14] and novel hologram representations [3, 15] remains an important direction for future work. Our code is available at [REVIEW].

2. RELATED WORK

Learned Computer-Generated Holography. Early learned approaches to single-color hologram synthesis primarily adopt U-Net–based Convolutional Neural Network (CNN) architectures [16] or resolution-preserving residual stacks to accelerate inference [1]. These methods require RGB-D inputs and are restricted to a fixed set of display-scene parameters. To remove the dependency on depth input, Liu et al. [17] introduce a multi-stage pipeline that separately performs depth estimation and RGB-D hologram optimization, resulting in increased computational cost. Ishii et al. [18] further extend this design with an additional hologram refinement stage, leading to even slower synthesis compared to prior methods [17, 19]. Akşit and Itoh [19] collapse these stages into a single-stage CNN for improved efficiency; however, the resulting 3d hologram exhibits inaccurate depth reconstruction. Concurrent works also explore feature distillation [20], diffractive decoding [21], propagation-adaptive CGH [22], and hologram synthesis from 2D-only inputs [23–25]. *However, they either focus on 2D holograms or assume fixed optical configurations, our work addresses these limitations by introducing a depth-input-free, single-stage approach that jointly supports display-scene parameters configuring for both single- and multi-color 3D holograms at interactive rates within single model.*

3. METHOD: CONFIGURABLE HOLOGRAPHY

Our method aims to synthesize 3D holograms for various holographic displays at inference time using a single learned model.

Our proposed method takes RGB-only images and display-scene parameters as inputs and outputs depths and holograms. *We leverage depth estimation as a beneficial parallel task to help generate accurate 3D holograms from 2D images. Thus, our model does not aim to compete the MDE models, rather our model could benefit from any potential advancements in their accuracy in the future.*

A. RQ1: Why is configurable CGH important and non-trivial?

Configurability is important for both users and researchers: practical holographic displays and viewing preferences require instant and frequent changes in propagation distance, volume depth, and brightness, while display development involves exploring wavelength and pixel pitch variations across hardware prototypes. However, achieving configurability is non-trivial because it requires a network to continuously adapt the *diffraction-driven* image formation process across parameters, rather than memorizing a fixed setting. This is particularly challenging for long-range light propagation in 3D hologram, where the optical field oscillates rapidly as light travels through space, and where different parameters exhibit uneven conditioning difficulty—an observation we analyze empirically later in Sec. A.

A.1. Problem Definition: Synthesizing 3D Holograms

Holographic displays rapidly play successive holograms to generate full-color images, which the Human Visual System (HVS) fuses into a perceived color reconstruction. Single-color holograms are computed for one wavelength; multi-color holograms are computed jointly for multiple primaries. Hologram synthesis can be modeled by the optimization

$$
\begin{aligned}
Z_0 = Z - \frac{VD}{2}, Z_n = Z + \frac{VD}{2}, \\
I_r(p,t,z) = \left| l_{(p,t)} e^{i \frac{\lambda_p}{\lambda_{p_{\text{anchor}}}} u_t} * h_p(\lambda_p, z, d_x) \right|^2, \\
\hat{u}_t, \hat{l}_{(p,t)} \leftarrow \underset{u_t, l_{(p,t)}}{\operatorname{argmin}} \mathcal{L}_{\text{img}} \left(\sum_{z=Z_0}^{Z_n} \sum_{p=1}^{P} \sum_{t=1}^{T} I_r(p,t,z), s I_{(p,z)} \right),
\end{aligned}
\tag{1}
$$

where $Z \in \mathbb{R}$ denotes the light propagation distance, $VD \in \mathbb{R}$ the volume depth, $P \in \mathbb{Z}$ the number of color primaries (i.e. typically three), and $p \in \mathbb{Z}$ the primary index. $T \in \mathbb{Z}$ denotes the number of subframes required to reproduce a full-color image (i.e. typically three), and $t \in \mathbb{Z}$ denotes the subframe index. $l_{(p,t)} \in \mathbb{R}^{P \times T}$ represents the light source power of the p-th primary at the t-th subframe. $\lambda_p \in$ 400–700nm denotes the active primary wavelength, while $\lambda_{p_{\text{anchor}}}$ denotes the anchor wavelength used for calibration. $u_t \in \mathbb{C}^{H \times W}$ represents the phase-only hologram, and d_x denotes the pixel pitch. Here, $I_{(p,z)} \in \mathbb{R}^{H \times W}$ denotes the target intensity image, $s \in \mathbb{R}$ controls brightness scaling, and $h_p \in \mathbb{C}^{H \times W}$ denotes the wavelength- and distance-dependent propagation kernel [26–28]. $\mathcal{L}_{\text{img}}$ measures the visual discrepancy between the target and reconstructed images. For single-color holography, l reduces to identity selection, whereas multi-color holography requires joint optimization across wavelengths. Existing optimization pipelines [6] typically require minutes per requested s and rely on RGB-D inputs. *Configurability* refers to replacing this parameterized optimization with a single model capable of adapting across a continuous range of (s, Z, VD, λ, d_x).

Diffraction exhibits scaling properties that relate wavelength, pixel pitch, and propagation distance: a hologram computed at one wavelength can approximate another by rescaling Z as $Z_2 = \frac{\lambda_2}{\lambda_1} \times Z_1$, and a similar relation connects pixel pitch and distance as $Z_2 = (\frac{p_1}{p_2})^2 \times Z_1$ (see Supplementary Sec. S9 for details and Point-Spread Function (PSF) visualizations). While these properties motivate shared structure across configurations, they hold only for single-color settings; multi-color holograms require jointly optimizing across wavelengths, where simple rescaling is insufficient. This partially explains why some parameters are harder to condition than others and motivates our explicit conditioning strategy. Additionally, iterative random-phase methods like GS and SGD [34, 35] are inherently non-configurable, as each run solves for a hologram under a fixed propagation kernel parameterized by (λ, d_x, Z). To verify this, we conduct an experiment in Supplementary Sec. S12, where we jointly optimize a hologram using SGD across four propagation distances. Our findings confirm that changing any parameter invalidates the accumulated phase updates, and joint optimization produces non-ideal phases due to conflicting gradients.

B. Answer to RQ1: A Configurable Model Prototype

We convert the parameterized hologram synthesis process in Eq. (1) into a single-stage learned model that is explicitly conditioned on display-scene parameters. Our final deployed model is a compact student model distilled from a stronger teacher using KD. This design targets configurability as *continuous generalization* over supported parameter ranges. Our model unifies three stages: (a) *wavefront synthesis*, mapping RGB inputs to a complex-valued field via a parameter-conditioned U-Net; (b) *wave propagation*, propagating the field via kernel; and (c) *phase extraction*, using imaginary part of the propagated field as hologram. We first introduce teacher, then the student model.

B.1. Teacher Model

We provide a complete layout of our teacher model in Fig. 2, the details of the modules can be found in Supplementary Sec. SA. The teacher takes I_{input}, λ_p, d_x, s, VD, and Z as inputs, denoted as $Param_{cond}$. During training, we vary these inputs so that the model can adapt the hologram synthesis process to preferred display–scene parameters at test time. Specifically, our training draws the input variables from

$$
\begin{aligned}
\lambda_p \subseteq \{(640, 515, 470)\}\ nm, s \subseteq \{1.0, 1.4, 1.8\}, \\
VD \subseteq \{4.0, 8.0\}\ mm, Z \subseteq \{2, 4, 7, 10\}\ mm, d_x \subseteq \{3.74\}\ \mu m.
\end{aligned}
\tag{2}
$$

Our choice of Z follows recent learned holography literature [1, 19, 36], and our VD choices cover common VR focal ranges (40–75 mm), roughly corresponding to placing virtual images from our VD between 5 Diopter to infinity. Given the scalability property in Sec. S9, we deliberately include a single pixel pitch in Eq. (2) to control training permutations and computational cost. To study broader conditioning over d_x (and other display parameters) with wider permutations, we also introduce an RGB-D condition variant (depth provided as input) and report its configurations and results in Tbl. 2 and Supplementary Sec. S17. Overall, we generate permutations of Eq. (2), resulting in 24 training cases, and train the teacher on the full set in one session.

Our teacher processes I_{input} using a U-Net structure [37] with encoder EfficientNet 1B [38] (code from [39]). We condition the decoder on the display-scene parameters and aggregate multi-scale decoder features with an FPN to form a shared latent code. We feed this latent code into three task-specific heads for predicting *phase-only holograms*, *light source powers*, and *depth* from RGB-only input. Their outputs are regularized by

$$
\mathcal{L}_{train} = \alpha_0 \mathcal{L}_{recon} + \alpha_1 \mathcal{L}_{light} + \alpha_2 \mathcal{L}_{depth}, \tag{3}
$$

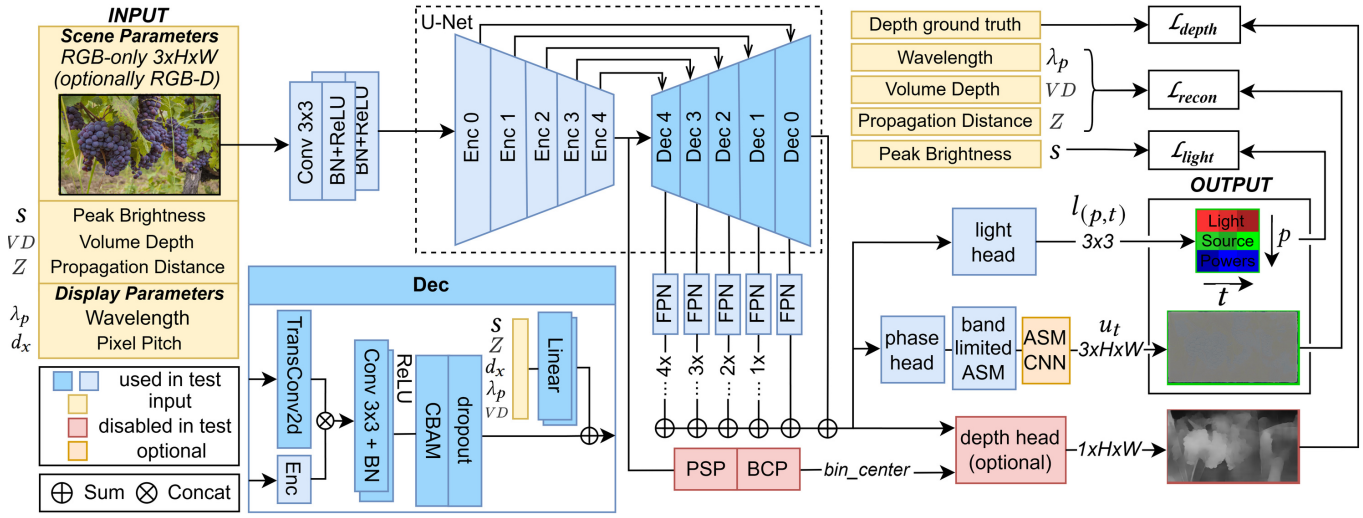


Fig. 2. Our teacher model. A Feature Pyramid Network (FPN) [29] connects to every decoder stage of our U-Net to leverage multi-scale spatial features from RGB-only or RGB-D inputs. Convolutional Block Attention Module (CBAM) [30] introduces channel and spatial attention, while each decoder stage is conditioned on peak brightness, wavelength, volume depth, propagation distance, and pixel pitch. A Pyramid Spatial Pooling (PSP) layer [31] aggregates global context for depth estimation, and a dedicated light head predicts the required light source powers. Here, *BCP* denotes the Bin Center Predictor (BCP) [32] (RGB-only source: [33]).

where $\alpha_0 = 1$, $\alpha_1 = 1$, $\alpha_2 = 30$ balance reconstruction, light power, and depth terms. $\mathcal{L}_{recon}$ is a multiplane reconstruction loss [5] that measures the discrepancy between the optically simulated reconstruction and the target image across depth planes. Given a ground-truth depth map, we generate target focal-stack images by applying per-pixel depth-dependent defocus following Kavaklı et al. [6]. $\mathcal{L}_{recon}$ combines three L_2 terms: a global reconstruction term, a masked term emphasizing in-focus regions, and a self-weighting term prioritizing high-intensity targets:

$$\mathcal{L}_{recon} = m_0 L_2(rec_k, target_k) + m_1 L_2(rec_k \cdot M_k, target_k \cdot M_k) + m_2 L_2(rec_k \cdot target_k, target_k \cdot target_k) + L_{smooth}, \quad (4)$$

where rec_k and $target_k$ denote the reconstructed and target images at depth plane k, M_k is the binary in-focus mask derived from the depth map, and L_{smooth} is a multi-scale total variation regularizer on the predicted phase [1, 6]. $\mathcal{L}_{light}$ constrains per-subframe laser powers to match the target color balance, following [6]. We also incorporate FLIP loss [40] to improve color consistency. Full mathematical definitions of all loss terms are provided in Supplementary Sec. S11.

Parameter Embedding. We propose to encode $Param_{cond}$ using novel scalar embeddings and a 1D PSF that captures the underlying physical conditions. The scalar display–scene parameters are embedded via sinusoidal encoding. A complex-valued 2D PSF parameterized by (λ, d_x, Z) is computed, from which the central x-axis is extracted as a 1D PSF. This 1D PSF is processed by two linear layers and concatenated with the scalars; the fused features are then passed through two additional linear layers to produce a conditioning vector injected into each decoder stage. Compared to prior 2D PSF conditioning [22, 41], our design is more efficient and uniquely supports multi-parameter conditioning. We include an ablation study in Supplementary Sec. S18 to validate the effectiveness of both branches.

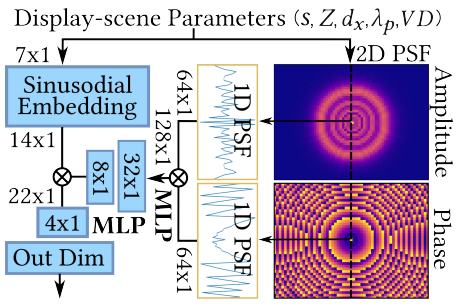


Fig. 3. Parameters Embedding layer.

Multi-task Learning. We adopt hard-parameter sharing [10, 42, 43], where a common U-Net backbone is shared across hologram, light power, and depth estimation tasks while maintaining task-specific heads. This design is motivated by the observation that prior single-stage RGB-only methods [19] produce highly inaccurate depth reconstruction; jointly learning MDE forces the shared representation to encode geometry relevant to 3D focus.

Phase-only Hologram Synthesis Task. Our phase head maps the latent code to a complex-valued field, which is propagated using a band-limited Angular Spectrum Method (ASM) with the input Z and λ across all subframes. We extract the imaginary part as the phase-only hologram. To support long propagation distances (e.g., $Z = 10\ mm$), we incorporate an ASM CNN block and skip-connect its output with the propagated phase. We find this block empirically necessary for long propagation; see Supplementary Sec. SA.1 for details.

Light Power Estimation Task. Our light head predicts a $t \times p$ (e.g. 3×3) matrix of light source powers in $[0, 1]$. It aggregates spatial information from the latent code and regresses intensities used to match the color of target reconstruction.

Depth Estimation Task. Besides phase and light heads, the teacher includes a depth head as an auxiliary task that improves hologram prediction when only RGB input is available. The head follows a bin-based formulation and combines encoder features with the latent code to predict dense depth. We define $\mathcal{L}_{depth} = \mathcal{L}_{silog} + \mathcal{L}_{gm} + \mathcal{L}_{tv}$, with Scale Invariant Log (SILog) [44], Gradient Matching (GM), and Total Variation (TV) terms detailed in Supplementary Sec. S11.

B.2. Student Model

While the teacher model achieves strong reconstruction quality, it requires 651 ms per frame on an NVIDIA A100 GPU at

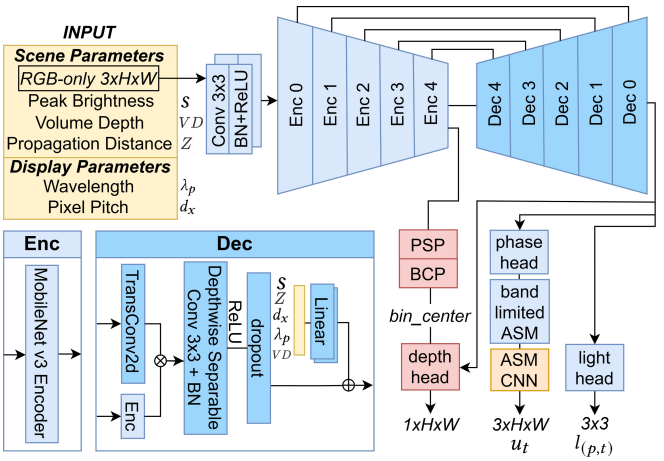


Fig. 4. Overview of our student model.

1920 × 1080 under fp32 (10.74M parameters), being inefficient. This motivates KD: rather than deploying the teacher directly or training a small model from scratch (which fails; see Sec. B.2), we distill the teacher's knowledge into a compact student (2.19M parameters) that achieves 39 ms inference time—a 16.7× speed-up over the teacher and 1.9× over Tensor V2 [1]—while preserving reconstruction quality. As shown in Fig. 4, we accelerate the teacher by converting it to a smaller student model using KD [9, 45, 46]. KD transfers knowledge from large teacher models to students, typically via response-based [47] or feature-based [48] distillation, and has been combined with Multi-task Learning (MTL) in other domains [49, 50]. *Without KD, smaller models trained independently fail to reliably solve our hologram synthesis task (see Sec. B.2).* The student takes the same inputs as the teacher and preserves the same outputs (phase, light, and depth), but replaces the encoder with MobileNetV3 [51] (code from [39]) and uses a lightweight decoder (e.g., depthwise separable convolutions). Conditioning embeddings and prediction heads are shared with the teacher; module-level descriptions are provided in Supplementary Sec. S17. The last stage of our student decoder conditions the U-Net with λ, s, d_x, VD, and Z. We apply logit-based KD [45] for our training

$$\mathcal{L}_{KD} = T^2 \cdot D_{KL}(S(y_{student}/T), S(y_{teacher}/T)), \quad (5)$$

where D_{KL} is the Kullback-Leibler divergence [52], S is softmax, and $T = 5$. We distill both phase and depth: $\mathcal{L}_{depth}$ targets the teacher-predicted depth, and we add a Charbonnier loss [53] to encourage smooth phase profiles. Our distillation loss is

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{KD}_{\text{phase}}} + \mathcal{L}_{\text{Charbonnier}} + \mathcal{L}_{\text{KD}_{\text{depth}}} + \mathcal{L}_{\text{depth}}. \quad (6)$$

Final loss uses equal weighting: $\mathcal{L} = \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{train}}$.

4. IMPLEMENTATION

We summarize key training and evaluation settings here, with full implementation details provided in the Supplementary material. Our training data are sampled from the Segment Anything (SA) dataset SA-1B [54], with ground-truth depth generated using MiDaS [55]. We select 44,000 images (0.4% of SA-1B) and synthesize target reconstructions following Kavaklı et al. [6]. All models are implemented in PyTorch [56] using Odak [57], and optimized with Adam [58] ($\beta_1 = 0.9$, $\beta_2 = 0.99$) and a CosineAnnealingLR [59] scheduler. We use a batch size of 16 and train both models for 20 epochs with an initial Learning Rate (LR) of 0.001. Owing to memory constraints, the teacher is trained at 800 × 800, while the student is trained at 1024 × 1024. All experiments are conducted on eight NVIDIA RTX 3090 GPUs, with a total training and distillation time of four days.

Training a single network to *continuously* adapt across multiple display–scene parameters is computationally intensive, and auxiliary depth supervision further increases training cost. To make this trade-off explicit, we (i) train a teacher–student pair to study configurability with RGB-only 3D hologram synthesis, and (ii) introduce an RGB-D condition variant (depth as input, no depth head) to enable broader conditioning over λ, d_x, and long-range Z with reduced training complexity.

5. EVALUATION

In this section, we aim to provide a comprehensive evaluation of our method and directly provide evidence to answer **RQ1** and **RQ2: Why are MDE and KD Beneficial to CGH?**

A. Evidence to RQ1: Quantitative and Qualitative results

Method	Input	PSNR↑	SSIM↑	LPIPS↓	FVVDP↑	Conf
Our Method (teacher)	RGB-only	27.4	0.91	0.42	8.0	Yes
Our Method (student)	RGB-only	27.4	0.91	0.41	7.9	Yes
Our Method (RGB-D)	RGB-D	28.2	0.93	**0.37**	**8.5**	Yes
HoloBeam [19]	RGB-D	25.6	0.88	0.44	8.2	No
Tensor V2 [1]	RGB-D	26.5	**0.94**	0.38	8.3	No
modified 3D NH [60]	RGB-D	**28.9**	0.91	0.42	8.2	No
Two-stage DepthAny V2	RGB-D	28.3	0.93	0.39	8.2	No

Table 1. Averaged quantitative comparison across methods.

Quantitative Analysis. As shown in Tbl. 1, our method achieves on par performance across all metrics. Although modified 3D NH attains the highest PSNR, Fig. 6 shows pronounced blurriness in its reconstructions, resulting in worse perceptual quality compared to our method. Our approach uniquely supports continuous configurability over display–scene parameters, whereas all baselines require costly retraining per configuration. Due to the large number of methods and parameter settings, full quantitative results (PSNR, SSIM, LPIPS [61], FLIP [40], and FVVDP [62]) and extended analyses are deferred to Supplementary Sec. SA. In summary, KD preserves fidelity: the student remains comparable to the teacher and existing learned CGH methods (△PSNR<0.1%) while retaining configurability. Notably, this marginal quality gap enables instant and continuous parameter adjustment at inference time, replacing costly per-configuration retraining (hours) and weight swapping (minutes). Consistent with iterative CGH pipelines [5, 6], increasing s and Z degrades image quality across methods; we quantify these effects and report statistical tests in Supplementary Sec. SB.

Ours	s	Z	VD	λ	d_x
RGB-only	*continuous*	*discrete*	*continuous*	*fixed*	*fixed*
RGB-D	*fixed*	*continuous*	*continuous*	*continuous*	*continuous*

Table 2. Configurability of our methods. *Continuous* denotes generalization to arbitrary values within the range at inference; *discrete* denotes a fixed set of trained values; *fixed* denotes a constant parameter. The RGB-only model prioritizes scene parameters (s, VD) with discrete Z, while the RGB-D model supports broader display parameters (λ, d_x, Z).

Configurability and Qualitative Evaluation. Tbl. 2 summarizes the flexibility of each model variant. Our RGB-only models continu-

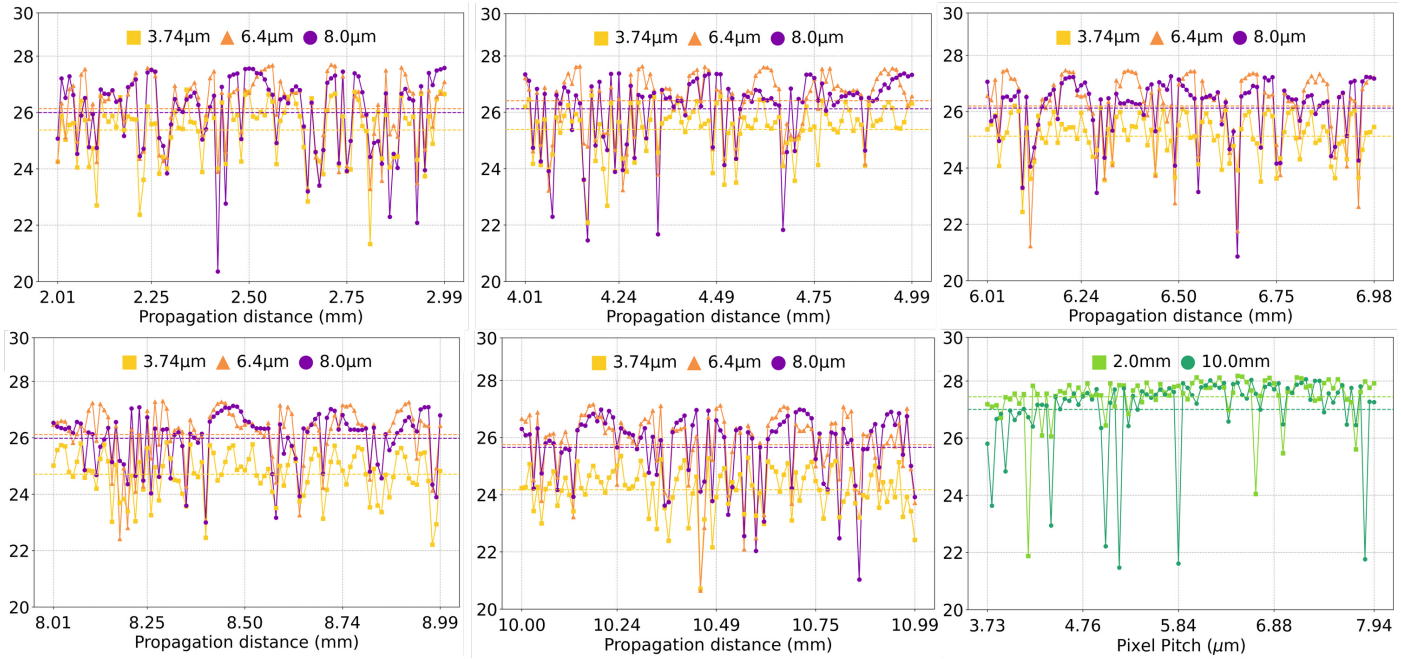


Fig. 5. PSNR (dB) of the RGB-D conditioned model on 100 DIV2K test images under novel (unseen) Z and d_x settings. Each point shows the mean PSNR at a randomly sampled (Z, VD) or (d_x, VD) pair. The top row and bottom-left panels evaluate a continuous 1 mm Z range at pixel pitches 3.74, 6.4, and 8.0 μm, while the bottom-right panel evaluates a continuous d_x range (3.7–8 μm) at $Z = 2.0$ and 10.0 mm. The model achieves an average PSNR of ≈26 dB with ≈1.1 dB standard deviation, with isolated drops (10–20%) due to limited model capacity across the wide conditioning range.

ously support s in $[1.0, 1.8]\times$ and VD in $[4, 8]$ mm, and support a discrete set of $Z \subseteq \{2, 4, 7, 10\}$ mm, while keeping d_x and λ fixed. This reflects the practical trade-off: *simultaneously covering more parameters continuously will increase training permutations and resource requirements significantly.* To probe broader display-parameter conditioning with reduced complexity, we use the RGB-D condition model and conduct three examples that support continuous ranges for Z, d_x, and λ:

- Z: continuous ranges of 2–3, 4–5, 6–7, 8–9, and 10–11 mm (total 5 mm) with three d_x values (3.74, 6.4, 8.0 μm).
- d_x: continuous range of 3.7–8.0 μm with two Z (2.0, 10.0 mm).
- λ: continuous ranges of 425–480, 510–565, 625–680 nm with two d_x (3.74, 8.0 μm) and two Z values (2.0, 10.0 mm).

All RGB-D training supports continuous VD in $[4, 8]$ mm with fixed s. Fig. 5 reports the PSNR distribution for long-range Z and d_x conditioning, with each point evaluated on the same 100 DIV2K images [64]. The full configuration experiments are included in Supplementary Sec. S19, where total of 3,364 novel (unseen) configurations is evaluated across Z, d_x, and λ. Across these studies, we observe that conditioning difficulty is not uniform: s and VD can be learned more smoothly, while long-range Z and display parameters (d_x, λ) require broader permutations and stronger inductive bias (e.g., our ASM CNN block for long Z). Overall, our model maintains consistent quality (average PSNR ≈ 26) with low variance (average std ≈ 1.1) across randomly generated parameters. Fig. 7 shows captured reconstructions from two of our three holographic display prototypes with different configurations, verifying our method under hardware setups; extra captured results are provided in Supplementary Sec. S16. Notably, our three prototypes span pixel pitches of 3.74, 6.4, and 8.0 μm, directly validating our configurability claim. Additionally, Fig. 6 compares reconstructions of our method, Tensor V2, and modified 3D NH across short and long Z. Compared with the other methods that requires retraining, our configurable model achieves competitive image quality while preserving correct focus/defocus cues across depth planes. To construct the modified 3D NH, we reimplement NH [60] and extend it to support RGB-D inputs.

B. RQ2: Why are MDE and KD Beneficial to CGH?

B.1. Depth Estimation

For RGB-only 3D hologram synthesis, depth ambiguity directly leads to incorrect focus/defocus cues across planes. Such errors are perceptually obvious even when image metrics like PSNR change only slightly. In particular, single-stage RGB-only learned CGH methods that do not estimate depth during training (e.g., Holobeam [19]) may produce plausible *2D* reconstructions but fail to generate 3D holograms with correct focus (see Fig. 8). *Our key insight is that jointly learning MDE and hologram synthesis enables depth-aware hologram generation from RGB-only inputs.* Specifically, the predicted monocular depth is normalized to the reconstruction range $Z \pm \frac{VD}{2}$, compressing the focal stack into the target volume. As in prior methods [1, 19, 36], we do not model eyepiece-to-diopter mapping, as it is orthogonal to SLM-plane synthesis. This makes our method applicable to photos, videos, and live streaming, where depth is unavailable.

Our depth estimation is intended as an auxiliary task for hologram synthesis, not as a replacement for dedicated MDE methods [66]. Our model is trained on 44K images, whereas state-of-the-art MDE methods use tens of millions. Despite this large gap, the depth head serves as a proof of concept that joint learning improves 3D hologram quality from RGB-only inputs. Therefore, we do not report standalone depth estimation metrics.

Training Data Strategy. Because CGH requires significantly less data than MDE to converge [1], our joint training must account

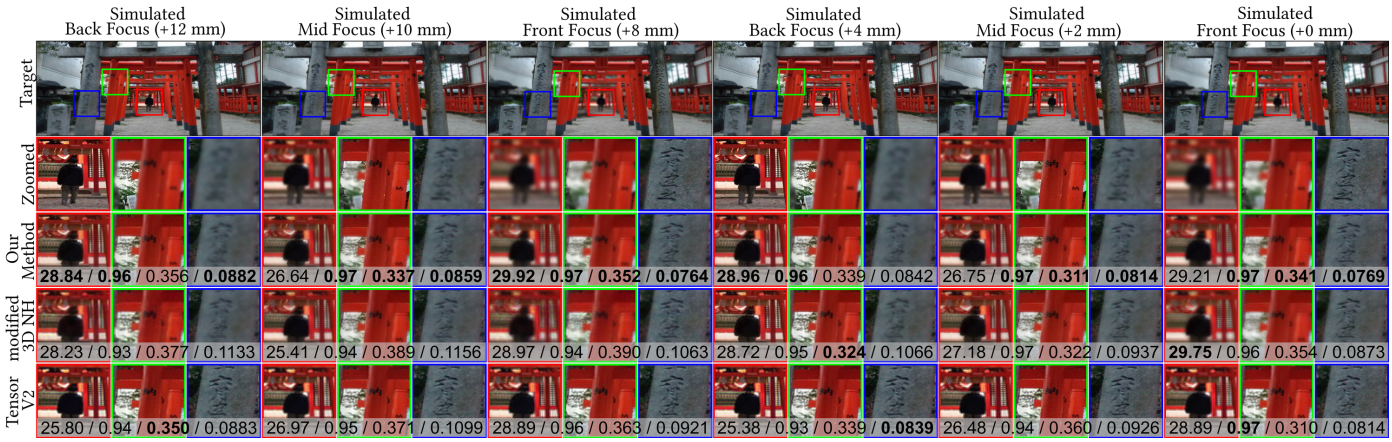


Fig. 6. Simulated reconstructions comparing our method, Tensor V2, and modified 3D NH for short and long propagation distances. Numbers report PSNR, SSIM, LPIPS, and FLIP, respectively (Source Image: [63]).

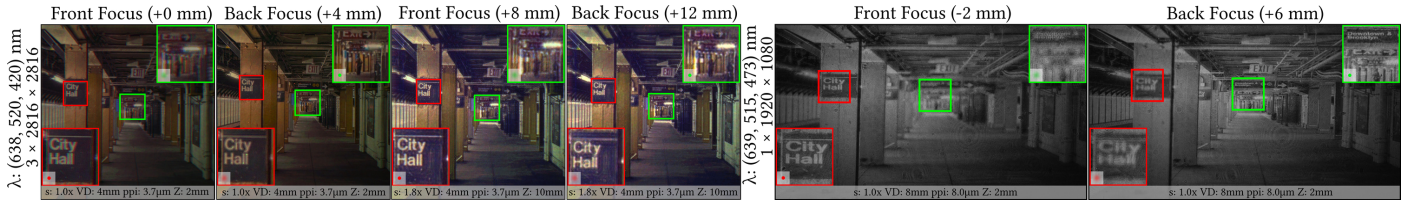


Fig. 7. Captured results from Jasper and Holoeye SLMs (2.1× pixel pitch difference) under varied parameters (source image: [65]).

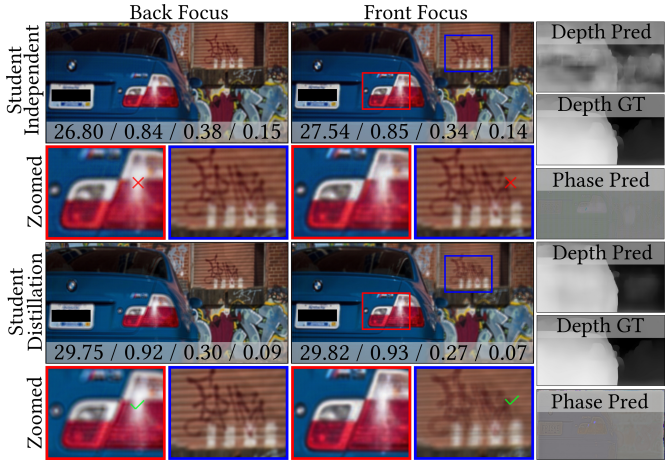


Fig. 8. Comparison of reconstructions, depth, and phase predictions between the independently trained student mode and the distilled student model. Numbers report PSNR, SSIM, LPIPS, and FLIP, respectively. (Source Image: [67])

for this imbalance. We use SA-1B [54] for its scale and resolution, generating depth pseudo-labels with MiDaS [55]. Our framework is agnostic to the specific depth estimator and can naturally benefit from future improvements; further discussion is provided in Supplementary Sec. S14.

B.2. Knowledge Distillation

KD helps us to create smaller and faster models for hologram synthesis that could not otherwise be trained effectively from scratch. Fig. 8 compares an independently trained student with our distilled student. When trained from scratch with the same setting as distillation, the independent student struggles to produce accurate depth maps, which leads to incorrect focus/defocus behavior and degraded reconstruction quality. In contrast, the distilled student demonstrates improved depth estimation, better color preservation, and higher image quality. Additional failure cases are provided in Supplementary Sec. S13.

Inference Time. Distilling the teacher into a student model results in a great improvement in speed. Fig. 9 compares inference time across models and resolutions. Our student model is consistently faster than all baselines, achieving 2× speed-up compared to Tensor V2 [1]. Specifically, our student model is 41%, 46%, and 44% faster than *Tensor V2*; and 17%, 14%, and 17% faster than *modified 3D NH*, respectively. The only method faster than our student is Holobeam [19]; however, as demonstrated in Fig. 8, excluding MDE as an auxiliary task leads to failures in reproducing correct focus cues in 3D hologram. Our student model can be further accelerated by removing the ASM CNN block; however, this component is essential for predicting holograms with Z larger than 4 *mm*. All timings are measured on an NVIDIA A100 40G GPU under fp32 with PyTorch.

6. RQ3: LIMITATION AND FUTURE WORK

Out-of-distribution Behavior. Extrapolation beyond the trained parameter ranges can degrade performance sharply (e.g., training over Z in 2–11 *mm* and evaluating at 12 *mm*). This limitation is relevant to training cost: expanding parameter range coverage by brute-force sampling becomes exponentially expensive. A core open problem for configurable holography is achieving robust generalization to novel configurations *without* densely training over them, especially for propagation distance under arbitrary volume depth in 3D hologram synthesis.

Defocus Accuracy. The defocus accuracy of reconstructions in our method remains strongly correlated with depth estimation

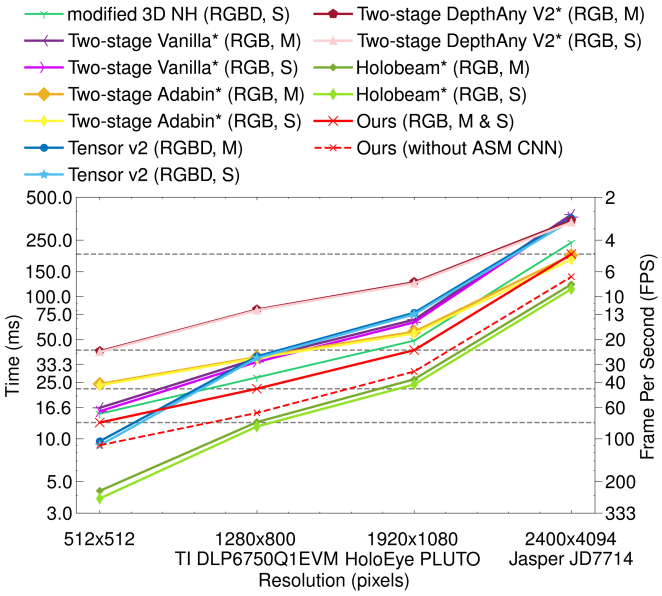


Fig. 9. Inference time comparison across models. *S* denotes single-color hologram and *M* denotes multi-color hologram synthesis using our light head module. *Two-stage* methods first estimate depth from RGB inputs and then generate 3D holograms. *Vanilla* uses U-Net* for both stages; *Adabin* employs Adabin [68]; and *DepthAny V2* uses DepthAnything V2 [66] for depth estimation. *Same architecture used in [19, 69–72].

accuracy; we provide error examples and their analysis in Supplementary Sec. S18. We emphasize that our depth estimation is introduced solely as an auxiliary signal to improve hologram synthesis, rather than to compete with dedicated MDE models such as DepthAnything V2 [66]; Accordingly, we do not include standalone depth estimation metrics. While the 3D accuracy in our method are greatly improved compared with Holobeam, the image quality remains slightly lower to RGB-D condition model. This gap is primarily attributable to training scale: our depth head is trained on only 44k images, whereas state-of-the-art MDE models typically rely on supervision from tens of millions of images. Future variants can therefore directly benefit from advances in MDE, enabling better geometric regularization and defocus in RGB-only 3D hologram synthesis.

Fluctuating Quality Across Conditioned Ranges. As shown in Fig. 5, we observe that reconstruction quality for 3D holograms can fluctuate as conditioning variables vary within a supported range (see Supplementary Sec. S19 for detailed analysis). The underlying cause is not yet fully understood. Compared with iterative methods [6], which optimize holograms under a fixed configuration with stable image quality, configurable holography must implicitly approximate a *global optimal solution* across a wide range of optical configurations. *Finding such a solution is inherently several orders of magnitude more complex than optimizing for a single configuration*, which partially explains the observed quality fluctuations. Narrowing the conditioning range reduces these fluctuations, as shown in Supplementary Sec. S19. Improving stability in feed-forward configurable methods may benefit from information driven inductive biases [73] or more carefully designed loss functions than naive per-pixel supervision [74].

Training Cost and Data Trade-offs. A single model that continuously adapts over multiple display-scene parameters is expensive to train, and incorporating auxiliary depth estimation further greatly increases data and computational requirements. As a result, we must trade off between (i) variety of configurability, (ii) range of configurability, and (iii) depth supervision diversity. In this paper, we trained a teacher-student pair as proof of concept to validate the feasibility of efficient and configurable RGB-only 3D hologram synthesis. Additionally, we train an RGB-D condition model (without a depth head) to demonstrate how far and wide configurability itself can extend under reduced computational complexity. We acknowledge that due to the limited resources, using network variants to validate different aspects of our approach is not ideal. The computational cost of dense training becomes prohibitive as parameter ranges expand: distilling a unified RGB-only student spanning 10 mm Z, 6 μm d_x, the visible spectrum, 10 mm VD, and 1–1.8× s would require 16 A100 GPUs and 20 training days due to exponentially growing parameter permutations. Accordingly, while our network is more efficient than existing learned CGH methods, it should be viewed as a prototype rather than the perfect solution for configurable holography. Future work should pursue unified architectures that jointly support broad configurability and depth-free inference, while avoiding dense parameter permutations through improved training and sampling strategies.

Different Hologram Types and Hardware Non-idealities. Our work assumes an ideal Spatial Light Modulator (SLM) with *smooth phase* and does not model non-idealities such as pixel fill-factor, crosstalk, higher-order diffraction, or SLM tilt. Although we provide extensive simulated results, broad hardware generalization and validation remain challenging because of the large number of combinations in the display hardware design. We therefore focus more on simulation and evaluate on three commonly used systems with similar optical architectures. Practical systems span a much wider design space—including wearable near-eye displays [75], waveguide combiners [76], and beam splitters—each introducing additional aberrations, stray light, and non-uniform degradation. At present, our framework does not yet support this extended design space. Further investigation is required for generalization across fundamentally different optical architectures, hardware calibration [60**?**], and configurable random-phase generation [74, 77] in future work.

7. CONCLUSION

We acknowledge that our research is based on well-established techniques, including networks, MTL, MDE, and KD. Our contribution lies not in them, but in (i) formulating *configurability* as a concrete objective for learned CGH, (ii) identifying which display–scene parameter can be continuously conditioned within a single model and which remain challenging, and (iii) providing, to our knowledge, the first empirical study quantifying the trade-offs between speed, parameter range, input requirements, conditioning difficulty, and image quality across thousands of novel configurations, (iv) revealing MDE as a beneficial auxiliary task for CGH. We believe these insights provide a useful reference for future configurable CGH research beyond any specific architecture. Our final goal is to build a CGH model that is continuously configurable and can adapt to novel configurations outside the training set without dense retraining. In other words, we seek a physically accurate CGH method that understands light propagation robustly across display and scene variations.

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. L. Shi, B. Li, and W. Matusik, "End-to-end learning of 3d phase-only holograms for holographic display," Light. Sci. & Appl. **11**, 247 (2022).
2. B. Spragg, "Colorful windmills shenzhen china." (2009).
3. D. Kim, S.-W. Nam, S. Choi, *et al.*, "Holographic parallax improves 3d perceptual realism," ACM Trans. on Graph. (TOG) **43**, 1–13 (2024).
4. D. Blinder, A. Ahar, S. Bettens, *et al.*, "Signal processing challenges for digital holographic video display systems," Signal Process. Image Commun. **70**, 114–130 (2019).
5. K. Kavaklı, Y. Itoh, H. Urey, and K. Akşit, "Realistic defocus blur for multiplane computer-generated holography," in *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR),* (IEEE, 2023), pp. 418–426.
6. K. Kavaklı, L. Shi, H. Urey, *et al.*, "Multi-color holograms improve brightness in holographic displays," in *ACM SIGGRAPH ASIA 2023 Conference Proceedings,* (ACM, Sydney, NSW, Australia, 2023), pp. –.
7. E. Markley, N. Matsuda, F. Schiffers, *et al.*, "Simultaneous color computer generated holography," in *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023,* J. Kim, M. C. Lin, and B. Bickel, eds. (ACM, 2023), pp. 22:1–22:11.
8. Y. Zhan, K. Kavaklı, H. Urey, *et al.*, "Autocolor: learned light power control for multi-color holograms," in *Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR) V,* vol. 12913 (SPIE, 2024), pp. 96–102.
9. L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," IEEE transactions on pattern analysis machine intelligence **44**, 3048–3068 (2021).
10. R. Caruana, "Multitask learning: A knowledge-based source of inductive bias1," in *Proceedings of the Tenth International Conference on Machine Learning,* (Citeseer, 1993), pp. 41–48.
11. L. Shi, F. Huang, W. Lopes, *et al.*, "Near-eye light field holographic rendering with spherical waves for wide field of view interactive 3d computer graphics," ACM Trans. Graph. **36**, 236:1–236:17 (2017).
12. C. Jang, K. Bang, M. Chae, *et al.*, "Waveguide holography for 3d augmented reality glasses," Nat. Commun. **15**, 66 (2024).
13. G. Kuo, F. Schiffers, D. Lanman, *et al.*, "Multisource holography," ACM Trans. Graph. **42** (2023).
14. M. Chae, K. Bang, D. Yoo, and Y. Jeong, "Étendue expansion in holographic near eye displays through sparse eye-box generation using lens array eyepiece," ACM Trans. Graph. **42**, 58:1–58:13 (2023).
15. S. Choi, M. Gopakumar, Y. Peng, *et al.*, "Time-multiplexed neural holography: A flexible framework for holographic near-eye displays with fast heavily-quantized spatial light modulators," in *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Vancouver, BC, Canada, August 7 - 11, 2022,* M. Nandigjav, N. J. Mitra, and A. Hertzmann, eds. (ACM, 2022), pp. 32:1–32:9.
16. N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," Ieee Access **9**, 82031–82057 (2021).
17. N. Liu, Z. Huang, Z. He, and L. Cao, "Dge-cnn: 2d-to-3d holographic display based on a depth gradient extracting module and zcnn network," Opt. Express **31**, 23867–23876 (2023).
18. Y. Ishii, F. Wang, H. Shiomi, *et al.*, "Multi-depth hologram generation from two-dimensional images by deep learning," Opt. Lasers Eng. **170**, 107758 (2023).
19. K. Akşit and Y. Itoh, "Holobeam: Paper-thin near-eye displays," in *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR),* (IEEE, 2023), pp. 581–591.
20. Y. Wang, W. Wan, J. Fu, and Y. Su, "Complex-valued attention feature distillation network for high-fidelity phase-only hologram generation," Opt. Express **33**, 25134–25145 (2025).
21. C. Isil, A. Chen, Y. Li, *et al.*, "Snapshot 3D image projection using a diffractive decoder," arXiv preprint arXiv:2512.20464 (2025).
22. N. Liu, K. Liu, Y. Yang, *et al.*, "Propagation-adaptive 4K computer-generated holography using physics-constrained spatial and Fourier neural operator," Nat. Commun. **16**, 7761 (2025).
23. C. Chang, B. Dai, D. Zhu, *et al.*, "From picture to 3D hologram: end-to-end learning of real-time 3D photorealistic hologram generation from 2D image input," Opt. Lett. **48**, 851–854 (2023).
24. C. Chang, C. Zhao, B. Dai, *et al.*, "Conversion of 2D picture to color 3D holography using end-to-end convolutional neural network," PhotoniX **6**, 30 (2025).
25. J. Sonker and M. M. Gore, "NetHolo: hologram reconstruction from RGB image in computational holography using deep neural network," The Imaging Sci. J. **73**, 962–970 (2025).
26. K. Matsushima and T. Shimobaba, "Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields," Opt. express **17**, 19662–19673 (2009).
27. K. Kavaklı, H. Urey, and K. Akşit, "Learned holographic light transport," Appl. Opt. **61**, B50–B55 (2022).
28. S. Choi, M. Gopakumar, Y. Peng, *et al.*, "Neural 3d holography: learning accurate wave propagation models for 3d holographic virtual and augmented reality displays," ACM Trans. on Graph. (TOG) **40**, 1–12 (2021).
29. T.-Y. Lin, P. Dollár, R. Girshick, *et al.*, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (Honolulu, HI, USA, 2017), pp. 936–944.
30. S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV),* (2018).
31. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Trans. Pattern Anal. Mach. Intell. **37**, 1904–1916 (2015).
32. A. Agarwal and C. Arora, "Attention attention everywhere: Monocular depth prediction with skip attention," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023,* (IEEE, Waikoloa, HI, USA, 2023), pp. 5850–5859.
33. B. I. Gashi, "Rahovec grapes and wine." (2012).
34. R. W. Gerchberg, "A practical algorithm for the determination of plane from image and diffraction pictures," Optik **35**, 237–246 (1972).
35. P. Chakravarthula, Y. Peng, J. Kollin, *et al.*, "Wirtinger holography for near-eye displays," in *ACM Trans. Graph. (SIGGRAPH Asia),* vol. 38 (2019).
36. L. Shi, B. Li, C. Kim, *et al.*, "Towards real-time photorealistic 3d holography with deep neural networks," Nature **591**, 234–239 (2021).
37. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III,* vol. 9351 of *Lecture Notes in Computer Science* N. Navab, J. Hornegger, W. M. W. III, and A. F. Frangi, eds. (Springer, 2015), pp. 234–241.
38. M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning,* (PMLR, 2019), pp. 6105–6114.
39. P. Iakubovskii, "Segmentation models pytorch," https://github.com/qubvel-org/segmentation_models.pytorch (2019).
40. P. Andersson, J. Nilsson, T. A. Moller, *et al.*, "Flip: A difference evaluator for alternating images," Proc. ACM on Comput. Graph. Interact. Tech. **3**, 15:1–15:23 (2020).
41. Y. Asano, K. Yamamoto, T. Fushimi, and Y. Ochiai, "Distance-adaptive unsupervised cnn model for computer-generated holography," in *ACM SIGGRAPH 2024 Posters,* (Association for Computing Machinery, New York, NY, USA, 2024), SIGGRAPH '24.
42. S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 33 (2019), pp. 4822–4829.
43. S. S. Sarwar, A. Ankit, and K. Roy, "Incremental learning in deep convolutional neural networks using partial network sharing," IEEE Access **8**, 4615–4628 (2019).
44. D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada,* Z. Ghahramani, M. Welling, C. Cortes, *et al.*, eds.

(NeurIPS, 2014), pp. 2366–2374.
45. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," NIPS 2014 Deep. Learn. Workshop (2015).
46. J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," Int. J. Comput. Vis. **129**, 1789–1819 (2021).
47. P. Micaelli and A. J. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," Adv. Neural Inf. Process. Syst. **32** (2019).
48. H. Chen, Y. Wang, C. Xu, *et al.*, "Learning student networks via feature embedding," IEEE Trans. on Neural Networks Learn. Syst. **32**, 25–35 (2020).
49. W.-H. Li and H. Bilen, "Knowledge distillation for multi-task learning," in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16,* (Springer, 2020), pp. 163–176.
50. A. Ignatov, G. Malivenko, D. Plowman, *et al.*, "Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* (2021), pp. 2545–2557.
51. A. Howard, M. Sandler, G. Chu, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV),* (2019).
52. S. Kullback, "Kullback-leibler divergence," (1951).
53. J. T. Barron, "A more general robust loss function," CoRR **abs/1701.03077** (2017).
54. A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision,* (2023), pp. 4015–4026.
55. K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," CoRR **abs/1907.01341** (2019).
56. A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," CoRR **abs/1912.01703** (2019).
57. K. Akşit, J. Beyazian, P. Chakravarthula, *et al.*, "Odak," (2024).
58. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings,* Y. Bengio and Y. LeCun, eds. (2015).
59. I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings,* (OpenReview.net, 2017).
60. Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," ACM Trans. on Graph. (TOG) **39**, 1–14 (2020).
61. R. Zhang, P. Isola, A. A. Efros, *et al.*, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018,* (Computer Vision Foundation / IEEE Computer Society, 2018), pp. 586–595.
62. R. K. Mantiuk, G. Denes, A. Chapiro, *et al.*, "Fovvideovdp: a visible difference predictor for wide field-of-view video," ACM Trans. Graph. **40**, 49:1–49:19 (2021).
63. Hirho, "Arakida-inari-jinja in suikyō-tenman-gū (15-4, tenjin 1-chōme, chūō-ku, fukuoka city, japan." (2022).
64. E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* (IEEE, Honolulu, HI, USA, 2017).
65. LogicalRailfan, "City hall north platform nb," (2023).
66. L. Yang, B. Kang, Z. Huang, *et al.*, "Depth anything v2," Adv. Neural Inf. Process. Syst. **37**, 21875–21911 (2024).
67. T. P. Father, "Laguna seca blue bmw m3," (2008).
68. S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021,* (Computer Vision Foundation / IEEE, 2021), pp. 4009–4018.
69. M. H. Eybposh, N. W. Caira, M. Atisa, *et al.*, "Deepcgh: 3d computer-generated holography using deep learning," Opt. Express **28**, 26636–26650 (2020).
70. H. Goi, K. Komuro, and T. Nomura, "Deep-learning-based binary hologram," Appl. Opt. **59**, 7103–7108 (2020).
71. J. Wu, K. Liu, X. Sui, and L. Cao, "High-speed computer-generated holography using an autoencoder-based deep neural network," Opt. Lett. **46**, 2908–2911 (2021).
72. R. Horisaki, Y. Nishizaki, K. Kitaguchi, *et al.*, "Three-dimensional deeply generated holography (invited)," Appl. Opt. **60**, A323–A328 (2021).
73. H. Pinkard, L. Kabuli, E. Markley, *et al.*, "Information-driven design of imaging systems," arXiv e-prints arXiv:2405.20559 (2024).
74. V. Chu, O. Pueyo-Ciutad, E. Tseng, *et al.*, "Artifact-resilient real-time holography," ACM Trans. Graph. **44** (2025).
75. J. Kim, M. Gopakumar, S. Choi, *et al.*, "Holographic glasses for virtual reality," in *ACM SIGGRAPH 2022 Conference Proceedings,* (Association for Computing Machinery, New York, NY, USA, 2022), SIGGRAPH '22.
76. C. Jang, K. Bang, M. Chae, *et al.*, "Waveguide holography for 3d augmented reality glasses," Nat. Commun. **15**, 66 (2024).
77. B. Chao, J. Yang, S. Choi, *et al.*, "Random-phase wave splatting of translucent primitives for computer-generated holography," arXiv preprint arXiv:2508.17480 (2025).
78. K. Liu, J. Wu, Z. He, and L. Cao, "4k-dmdnet: diffraction model-driven network for 4k computer-generated holography," OptoElectron Adv **6**, 220135–1–220135–13 (2023).
79. C.-K. Hsueh and A. A. Sawchuk, "Computer-generated double-phase holograms," Appl. optics **17**, 3874–3883 (1978).
80. A. Agarwal and C. Arora, "PixelFormer: Attention attention everywhere: Monocular depth prediction with skip attention," https://github.com/ashutosh1807/PixelFormer (2022).
81. B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, *et al.*, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," ACM Trans. on Graph. (TOG) (2019).
82. V. F. Catering, "Pumpkin pie cupcake with whipped kreme and candy corn gummy," (2009).
83. schermpeter42, "Yellow, red and blue," (2014).
84. kennejima, "Guy fieris vegas kitchen & bar," (2014).
85. P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV,* (2012).
86. A. Ross and V. L. Willson, "One-way anova," in *Basic and advanced statistical tests,* (Brill, Leiden, Netherlands, 2017), pp. 21–24.
87. Hirho, "Ōmiya view south from north end Ōmiya-1-chōme chūō-ku fukuoka city 20230706." (2023).
88. E. Hunt, "Fruits and vegetables at pike place market." (2003).

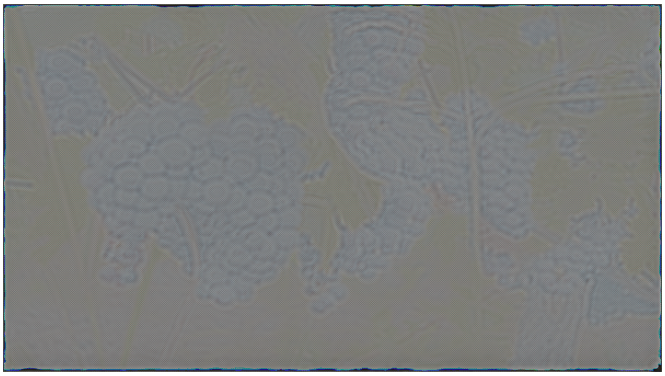


Fig. 10. The example of our phase-only hologram.

SUPPLEMENTARY 1

8. PHASE PROFILE AND EXPERIMENTAL SETUP

A. Phase Image Example

Fig. S10 shows the example phase profile our model predicted.

B. Hardware Image

Fig. S11a and Fig. S11b show the photographs of two of the three holographic display prototypes we used in this paper. The optical path of our display prototype begins with a laser light source (LASOS MCS4), which integrates three individual laser lines. The emitted light from a single-mode fibre is collimated using a Thorlabs LA1708-A plano-convex lens with a 200 mm focal length. This linearly polarized, collimated beam is then directed by a beamsplitter (Thorlabs BP245B1) toward our phase-only SLM, the Holoeye Pluto-VIS (1920×1080 px, 8 μm), Holoeye LETO (1920×1080 px, 6.4 μm), or Jasper JD7714 (2400×4094, 3.74 μm). The modulated beam subsequently passes through a lens system comprising Thorlabs LA1908-A and LB1056-A, with focal lengths of 500 mm and 250 mm, respectively. Following this, a pinhole aperture (Thorlabs SM1D12) is positioned at the focal plane of the lenses. Finally, we capture the holographic reconstructions using a lensless image sensor (Point Grey GS3-U3-23S6M-C USB 3.0), which is mounted on an X-stage (Thorlabs PT1/M) with a travel range of 0 to 25 mm and a positioning precision of 0.01 mm.

9. DIFFRACTION'S SCALABILITY PROPERTY

A hologram computed for one wavelength can be reused for another by scaling the propagation distance as $Z_2 = \frac{\lambda_2}{\lambda_1} \times Z_1$. Fig. S12 illustrates this property: the two PSF patterns are visually similar despite different (Z, λ) pairs, because the Fresnel number is preserved under this scaling. A related scaling connects pixel pitch and propagation distance as $Z_2 = (\frac{p_1}{p_2})^2 \times Z_1$. These properties apply to *single-color* settings where a fixed-wavelength model can be reused across wavelengths by adjusting Z. However, they do not resolve multi-color cases (Eq. 2 in the main paper), where wavelengths interact through a joint optimization objective and a single rescaling factor is insufficient. Given the scalability property, we deliberately include a single pixel pitch in our RGB-only training variable set to control training permutations and computational cost, while using the RGB-D condition variant to study broader pixel pitch conditioning.

10. MODEL

A. Model Structure

A.1. Teacher Model

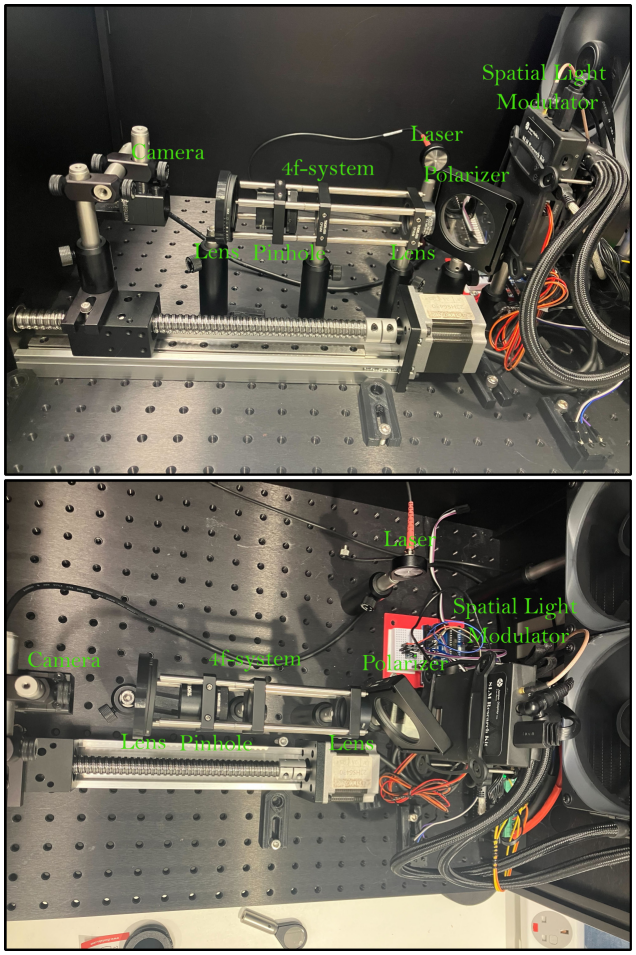
Tbl. S**??** summarizes the architecture details of both the teacher and student models.

FPN Structure Following the U-Net decoder, we aggregate features from multiple decoder stages using an FPN [29]. Each decoder output D_i is fed into its corresponding FPN layers according to their level. Each FPN layer in our model consists of a convolutional layer, batch normalization, and a nonlinear activation function ReLU followed by a bilinear upsample layer and an additional convolutional layer. Each FPN layer upsamples feature map by a scale of 2. Each decoder output is projected to a common channel dimension and progressively upsampled to the full resolution. To upsample every decoder output to the same scale, the D_i will be processed iteratively by FPN layers. For example, consider the first decoder output $Dec4$ of size $D_4 \text{x} \frac{H}{32} \text{x} \frac{W}{32}$, the $Dec4$ will be processed by FPN layers five times. Each time the resolution is doubled, resulting in an output that is the same size $D_0 \text{x} H \text{x} W$. The aligned features are then fused (summation after lightweight convolutions) to obtain a single latent code that is shared by all prediction heads.

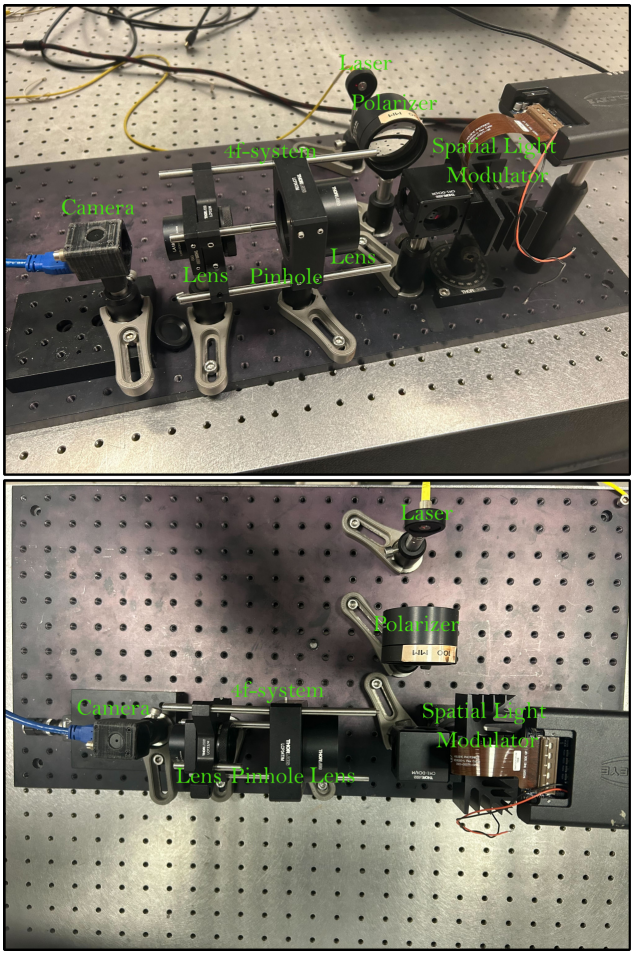
Phase Prediction Layer Structure The phase prediction head is a Conv2d layer with no activation function, allowing the model to infer phase values without explicit constraints. Following the phase head are bandlimited ASM kernel and ASM CNN. We found that our model will only converge when extra learnable parameters are provided after the bandlimited ASM method during training, which makes ASM CNN essential for long propagation distance estimation. The ASM CNN contains a convolutional layer, followed by three consecutive convolutional layers, batch normalization, and ReLU. The phase head predicts a complex-valued field and uses band-limited ASM, with an additional refinement block for long propagation distances. A similar structure is also used in Tensor V2, where the second-stage network is used after free space propagation. This observation highlights that long propagation distance prediction is inherently more challenging and requires extra prior as part of the network to adapt to the long-distance light propagation better. Please also note that our model only supports direct phase encoding.

Light Prediction Layer Structure The light prediction layer aggregates information in the model prediction *out* and predicts light intensity. The light head aggregates spatial features and predicts per-(subframe, color primary) light powers constrained to $[0, 1]$. Given the output of the model of size $D_0 \text{x} H \text{x} W$, first, the convolutional layers will downsample it by a scale of 4. Then, an adaptive pooling operation is applied to reduce the spatial dimensions to D_0x1x1. The pooled data will be fed into two linear layers, which manipulate the channel dimension to yield a value of 9, resulting in a 9x1x1 feature map. The final light intensity prediction *laser* is obtained by reshaping this output to a 1x3x3 matrix. This design ensures an efficient and accurate prediction of light intensity from the decoder output. Since the data range of light intensity is between 0 and 1, Sigmoid is used to ensure the model output is appropriately scaled.

Depth Prediction Layer Structure The depth prediction layer consists of a BCP and a depth head, following the PixelFormer implementation [80]. It adopts a bin-based formulation with



(a) First holographic display prototype: Jasper JD7714.



(b) Second holographic display prototype: Holoeye Pluto-VIS.

Fig. 11. Hardware prototypes used in evaluation: (a) Jasper JD7714 and (b) Holoeye Pluto-VIS.

	Input	Conf	Speed	Hologram			SLM Refresh Rate	Stage	3D	Depth Accuracy	Learned	Max Z (mm)	VD (mm)
				Type	PD	DP							
Our Method	RGB-only	Yes	Fast	M+S	8	No	60 Hz	Single	True	Moderate	Full	~10.0	~8.0
HoloBeam[19]	G-only	No	Fast	S	8	Yes	60 Hz	Single	True	Inaccurate	Full	~0.0	~6.0
Multi-DNN [18]	RGB-only	No	Slow	S	8	Yes	60 Hz	Three	True	Moderate	Full	~50.0	~2.0
NH [60]	RGB-only	No	Fast	S	8	No	60 Hz	Two	False	Accurate	Full	~100.0	~0.0
NH3D [?]	RGB-D	No	Slow	S	8	No	60 Hz	Two	True	Accurate	Semi	~8.2	~4.4
TensorV2 [1]	RGB-D	No	Fast	S	8	Yes	60 Hz	Two	True	Accurate	Full	~12.0	~6.0
DGE-CNN [17]	RGB-D	No	Slow	S	8	No	58 Hz	Two	True	Moderate	Semi	~10.0	~30.0
4K-DMDNet [78]	RGB-D	No	Slow	S	8	No	58 Hz	Two	False	Accurate	Full	~300.0	~0.0
Time-multiplexed [15]	RGB-D	No	Slow	S	4	No	480 Hz	Two	True	Accurate	Semi	~79.0	~12.0

Table 3. Comparison of hologram synthesis methods. Our method generates both single-color (S) and multi-color (M) holograms from an RGB-only input for a preferred set of display-scene parameters. In *Input*, G-only denotes green-channel-only. In *Speed*, Fast and Slow denote $>$ 10 FPS and $<$ 10 FPS, respectively, at 1920 $\times$ 1080 resolution. In *Learned*, *Full* denotes a fully learning-based method, while *Semi* denotes a hybrid of learning and optimization. NH [60] is fully learning-based, whereas NH3D [28] uses learning only for hologram refinement. *Conf*, *PD*, *DP*, Z, VD, and *SLM* denote Configurable, pixel depth, double phase encoding [79], maximum propagation distance, scene volume depth, and Spatial Light Modulator, respectively.

PSP and a bin-center predictor. *Bin Center Predictor:* Given a feature map E_4 of size $E_4 \times \frac{H}{32} \times \frac{W}{32}$, PSP applies adaptive global pooling at scales $\{1, 2, 3, 6\}$. The pooled feature is then passed to the BCP, following [32]. By predicting bin-center maps,

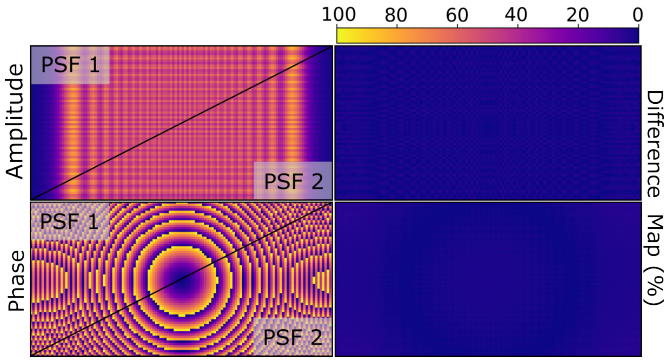


Fig. 12. Diffraction scalability. Scaling propagation distance by the wavelength ratio produces similar PSF patterns. PSF 1: $Z = 3\ mm$, $\lambda = 515\ nm$; PSF 2: $Z = 3.723\ mm$, $\lambda = 416\ nm$. Both use $d_x = 3.74\ \mu m$. The amplitude and phase remain highly consistent, while the difference maps on the right stay close to zero, showing that the main diffraction structure is the same.

Teacher			
Module	**Layer**	**Ch.**	**Match**
Enc. *EffNet-b1* *6.51M*	#0	16→32	Dec #0
	#1	32→24	Dec #1
	#2	24→40	Dec #2
	#3	40→112	Dec #3
	#4	112→320	Dec #4
Dec.+FPN *4.17M*	#4	320→272	Enc #4
	#3	272→176	Enc #3
	#2	176→112	Enc #2
	#1	112→88	Enc #1
	#0	88→60	Enc #0
Heads	Phase	60→3	–
	Depth	60→1	–
	Laser	60→9	–

Table 4. Teacher model.

Student			
Module	**Layer**	**Ch.**	**Match**
Enc. *MBV3-S* *0.93M*	#0	16→16	Dec #0
	#1	16→16	Dec #1
	#2	16→24	Dec #2
	#3	24→48	Dec #3
	#4	48→576	Dec #4
Dec. *1.2M*	#4	576→336	Enc #4
	#3	336→192	Enc #3
	#2	192→112	Enc #2
	#1	112→72	Enc #1
	#0	72→52	Enc #0
Heads	Phase	52→3	–
	Depth	52→1	–
	Laser	52→9	–

Table 5. Student model.

BCP reformulates depth estimation from pure regression into a classification-regression task. *Depth Prediction Head:* Given a feature map D_4 of size $D_4 \times \frac{H}{32} \times \frac{W}{32}$, the depth head takes the latent feature and BCP output, then multiplies them along the channel dimension to produce the final depth.

11. LOSS FUNCTIONS

Our training objective combines four loss components: flip loss, reconstruction loss, light loss, and depth loss. We provide the mathematical formulations below.

A. Flip Loss

We employ FLIP loss [40] to improve color accuracy by penalizing perceptual color differences in IPT space. We simplify the original formulation by setting $\Delta feature = 0$:

$$\mathcal{L}_{flip} = \sum_i \Delta color(I_{p_i}, I_{t_i}), \tag{7}$$

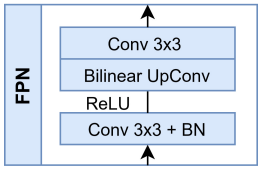


Fig. 13. Overview of the FPN aggregation module.

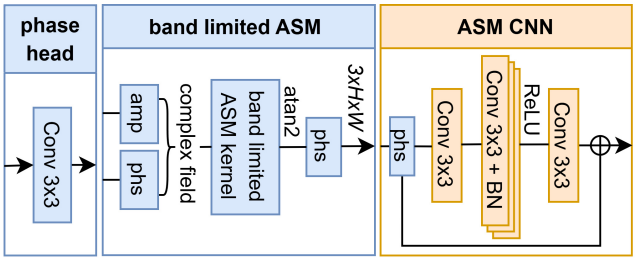


Fig. 14. Overview of the phase head module.

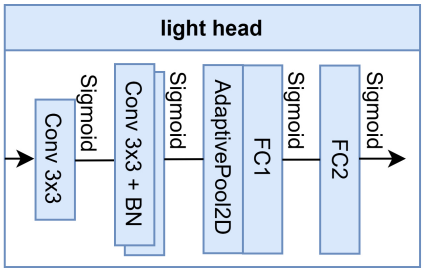


Fig. 15. Overview of the light head module.

where $\Delta color$ computes the redistributed Euclidean distance in IPT color space between predicted I_p and target I_t.

B. Reconstruction Loss

The reconstruction loss $\mathcal{L}_{recon}$ is shown below:

$$\begin{aligned}\mathcal{L}_{recon} = {} & m_0 L_2(rec_k, target_k) + m_1 L_2(rec_k * mask_k, target_k * mask_k) \\ & + m_2 L_2(rec_k * target_k, target_k * target_k) + L_s mooth, \end{aligned} \tag{8}$$

where m_0, m_1, and m_2 represent weights, rec_k, $mask_k$ and $target_k$ represent the reconstructed image, binary mask, and target image at the kth plane. Weighted with m_0, $L_2(rec_k, target_k)$ is the L2 norm that evaluates the entire image with respect to a target image, $m_1 * L_2(rec_k * mask_k, target_k * mask_k)$ applies a mask to the reconstructed and target images before computing the L2 norm. Finally, the $m_2 * L_2(rec_k * target_k, target_k * target_k)$ multiplies the reconstructed and target images together before computing the L2 norm, which emphasizes the regions of the target image that have high values. Additionally, $L_s mooth$ refers to phase smoothing loss, we used multi-scale TV loss introduced by [6] to ensure the smoothness in phase prediction, and regularization loss introduced by [1] to constrain the standard deviation and mean value of phase prediction to be close to 0.

C. Light Loss

Following [6], we constrain laser intensity per frame through four regularization terms. Let $channel_{sum}[i] = \sum_c laser[i, c]$ for frame $i \in \{0, 1, 2\}$:

$$\mathcal{L}_{mean} = \frac{1}{N} \sum_i (channel_{sum}[i] - peak_{amplitude})^2, \tag{9}$$

$$\mathcal{L}_{abs} = \frac{1}{N} \sum_i \left| channel_{sum}[i] - peak_{amplitude} \right|, \tag{10}$$

$$\mathcal{L}_{amax} = \sum_j (\max(recons[j]) - \textstyle\sum_{dim=0}(peak_{amplitude}))^2, \tag{11}$$

$$\mathcal{L}_{rgb} = \sum_{c,i} (\max(peak_{amplitude}[c]) - peak_{amplitude}[c, i]), \tag{12}$$

where j indexes depth frames, c indexes color primaries, and i indexes subframes. $\mathcal{L}_{mean}$ and $\mathcal{L}_{abs}$ penalize deviations of per-frame channel sums from the target amplitude; $\mathcal{L}_{amax}$ constrains maximum reconstructed intensity; and $\mathcal{L}_{rgb}$ encourages the three primaries to follow the R-G-B pattern. The combined light loss is $\mathcal{L}_{light} = \gamma(\mathcal{L}_{mean} + \mathcal{L}_{abs} + \mathcal{L}_{rgb} + \mathcal{L}_{amax})$ with $\gamma = 1 \times 10^5$.

D. Depth Loss

The depth loss function consists of three sub-loss functions: $\mathcal{L}_{silog}$, $\mathcal{L}_{gm}$, and $\mathcal{L}_{tv}$. $\mathcal{L}_{silog}$ represents the scale-invariant loss invented by eigen et al. [44], $\mathcal{L}_{gm}$ represents the gradient matching loss, which compares edges of estimated depths with ground truth depth maps, and $\mathcal{L}_{tv}$ represents the total variant loss, which smoothes the edge of objects in the depth map.

$\mathcal{L}_{silog}$: Given predicted depth $\widehat{d}_i$ and ground truth d_i^* at ith pixel, the logarithmic distance between $log(\widehat{d}_i)$ and $log(d_i^*)$ is calculated as $D_i = log(\widehat{d}_i) - log(d_i^*)$, calculated as:

$$\mathcal{L}_{silog} = \alpha\left(\frac{1}{n}\sum_i (D_i)^2 - \frac{\lambda}{n^2}\left(\sum_i D_i\right)^2\right) \tag{13}$$

$\mathcal{L}_{gm}$: We compute image gradients using the Sobel operator in horizontal (x) and vertical (y) directions. For predicted depth $\widehat{d}$ and ground truth d^*, gradient magnitudes are:

$$\widehat{G} = \sqrt{\widehat{G}_x^2 + \widehat{G}_y^2}, \quad G^* = \sqrt{G_x^{*2} + G_y^{*2}}, \tag{14}$$

where $\widehat{G}_x$, $\widehat{G}_y$ are gradients of $\widehat{d}$, and G_x^*, G_y^* are gradients of d^*. The gradient matching loss is:

$$\mathcal{L}_{gm} = \beta\frac{1}{n}\sum_i (\widehat{G}_i - G_i^*)^2, \tag{15}$$

where β is the weight and n is the total number of pixels.

$\mathcal{L}_{tv}$: Given the predicted depth $\widehat{d}$, we first compute the adjacent gradients in both the x and y directions g_x and g_y:

$$\text{g\_x} = \widehat{d}[:,:,1:,:] - \widehat{d}[:,:,:-1,:] \tag{16}$$

$$\text{g\_y} = \widehat{d}[:,:,:,1:] - \widehat{d}[:,:,:,:-1] \tag{17}$$

The edge-smoothing loss $\mathcal{L}_{tv}$ is defined as the sum of the mean absolute values of these gradients, where n denotes the total number of pixels.

$$\mathcal{L}_{tv} = \frac{1}{n}\sum_i |\text{g\_x\_i}| + \frac{1}{n}\sum_i |\text{g\_y\_i}| \tag{18}$$

The final $\mathcal{L}_{depth}$ is defined as the sum of the $\mathcal{L}_{silog}$, $\mathcal{L}_{gm}$ and $\mathcal{L}_{tv}$.

$$\mathcal{L}_{depth} = \mathcal{L}_{silog} + \mathcal{L}_{gm} + \mathcal{L}_{tv} \tag{19}$$

12. NON-CONFIGURABILITY OF ITERATIVE METHODS

Iterative hologram optimization solves for a phase-only hologram under a fixed propagation kernel. To test whether such methods can be extended to support multiple configurations simultaneously, we optimize either: (i) a hologram with SGD under a single fixed propagation distance, or (ii) a single hologram jointly over four propagation distances $Z \in \{1, 2, 3, 4\}$ mm by summing the reconstruction losses across all four settings.

Fig. S16 shows the resulting reconstructions and holograms. Panel (a) presents reconstructions obtained by SGD with a fixed Z. The optimized hologram yields clean results at the target setting, with correct front/back focus behavior. Panel (b) presents reconstructions obtained when SGD jointly optimizes a single hologram over four different Z values. In this case, the reconstructions at all distances exhibit severe noise and loss of correct focus, indicating strong interference among the inconsistent optimization objectives. Panel (c) compares the corresponding holograms from (a) and (b): the hologram optimized for multiple Z values is visibly corrupted with fringes, which explains the degraded reconstructions across all focal settings.

These results show that iterative methods cannot produce a single shared hologram that remains valid across different propagation configurations. The phase updates required by one propagation kernel conflict with those required by others, so changing the configuration necessitates re-optimizing the hologram from scratch. This makes iterative optimization fundamentally non-configurable.

13. KD AND CGH ANALYSIS

A. Failure Cases Of Individually Trained Student Model

Fig. S17 compares the independently trained student with the distilled student. Although both use the same training setting, the independently trained student produces less accurate depth maps, leading to weaker focus/defocus effects and lower reconstruction quality. By contrast, the distilled student achieves better depth estimation, color preservation, and overall image quality.

14. TRAINING DATA STRATEGY

Unlike depth estimation, which requires extensive training data to learn robust geometric priors across diverse scenes, CGH is fundamentally a mapping task that reflects diffraction physics. Consequently, CGH networks exhibit significantly lower data requirements for convergence and generalization. Prior work [1] has demonstrated that typical learned CGH models can achieve robust performance with only around 1,000 training images, as the primary learning objective is to capture the wavelength- and distance-dependent light transport kernel rather than high-level scene understanding. Given this asymmetry, our joint learning framework faces a design trade-off: while CGH alone requires less data, incorporating MDE as an auxiliary task demands much larger and more diverse training data. To address this, we select the Segment Anything dataset SA-1B [54], which provides two advantages. First, SA-1B contains tens of millions of high-resolution images with diverse content, satisfying the data scale and variability required for effective depth estimation. Second, unlike common depth datasets that are limited to lower resolutions (e.g. NYU Depth V2 [85] at 640×480), SA-1B provides high-resolution images suitable for hologram synthesis training. For ground-truth depth supervision, we employ MiDaS [55] to generate pseudo-labels. We acknowledge that MiDaS is not the state-of-the-art (SOTA) in MDE and exhibits known limitations, including scale ambiguity and reduced accuracy in complex scenes. However, as our goal is to demonstrate the *feasibility* of depth-assisted RGB-only 3D hologram synthesis rather than advancing MDE itself, MiDaS provides sufficient supervision quality for this proof of concept. Our framework is independent of the specific depth estimator used for pseudo-labeling and naturally inherits improvements from future depth models.

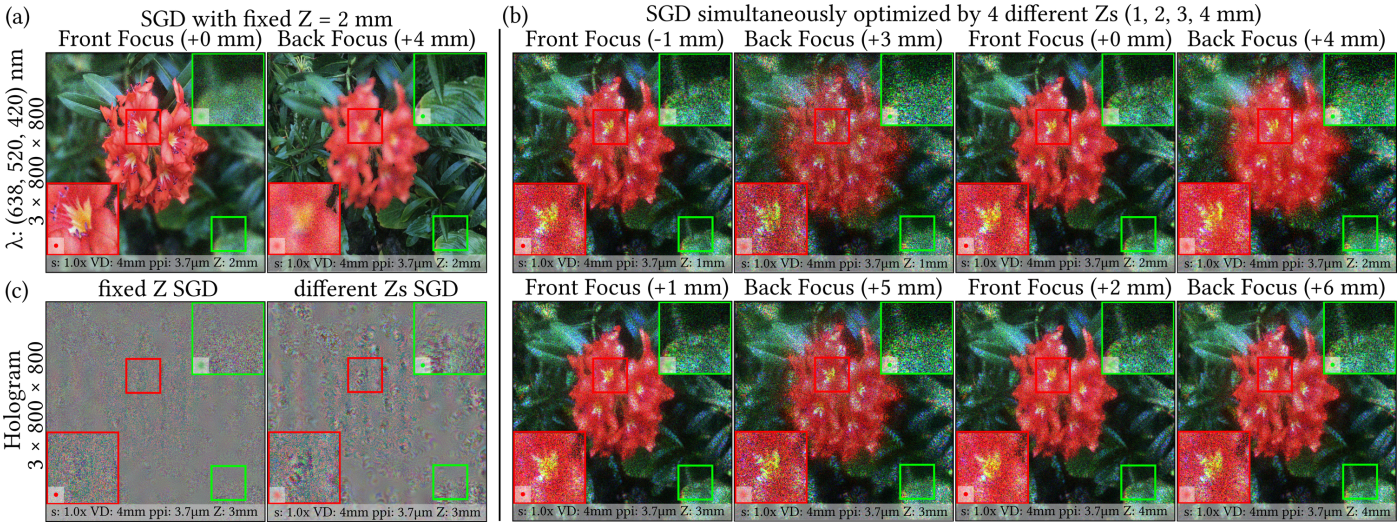


Fig. 16. Empirical evidence that SGD-based hologram optimization is non-configurable. (a) Reconstructions from SGD with a fixed Z. (b) Reconstructions from jointly optimizing one hologram over four propagation distances ($Z = 1, 2, 3, 4$ mm), causing severe artifacts at all settings. (c) Corresponding holograms from (a) and (b) (Source Image: [81]).

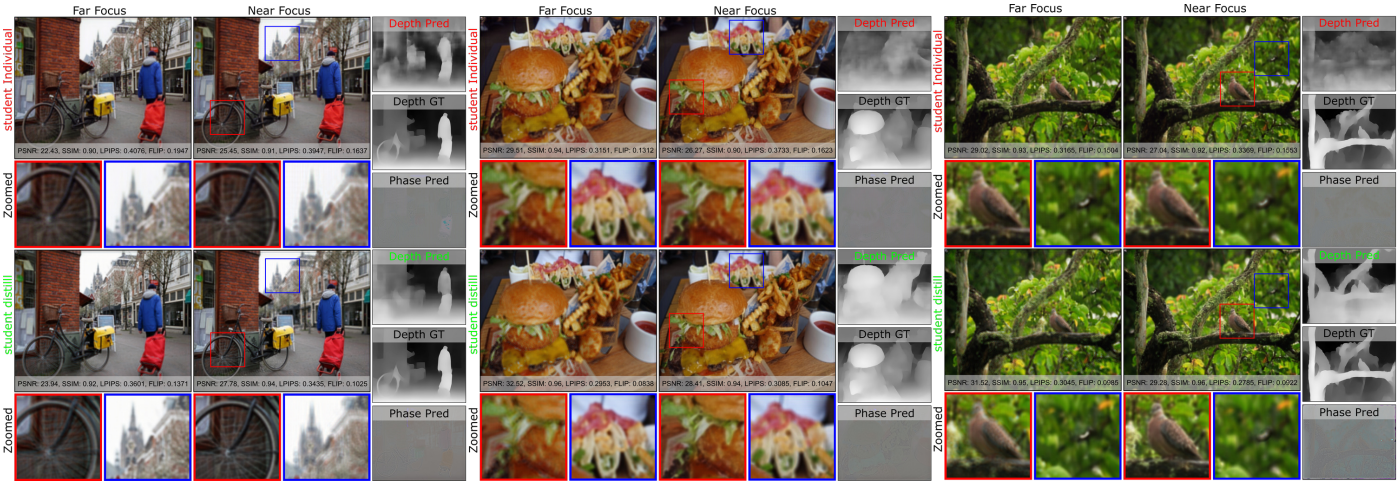


Fig. 17. Comparison of reconstructions, phase, and depth between the independently trained student model (in red) and the distilled student model (in green). From left to right, top to bottom: (Source Image: [82], [83], and [84])

15. EXTENDED EVALUATION TABLE

Tbl. 6 provides the complete quantitative evaluation at different holography settings, including all conditions supported by our teacher and student models.

A. Quantitative Analysis

We compute PSNR, SSIM, LPIPS [61], FLIP [40], and FVVDP [62] using 100 test images from DIV2K [64] not used during training. The student model maintains comparable performance to the teacher ($\triangle$PSNR$<$ 0.1%), supporting the practical value of distillation. Across configurations, PSNR and SSIM vary by 7.0% and 6.3% between best and worst settings, while remaining competitive with fixed-configuration baselines (+4.9% PSNR, +2.9% SSIM relative to per-configuration models). Consistent with prior iterative pipelines [5, 6], increasing s or Z reduces quality; for example, $s = 1.8\times$ induces decreases of 1.9% (PSNR), 1.1% (SSIM), 2.6% (LPIPS), 8.3% (FLIP), and 3.6% (FVVDP).

16. HARDWARE CAPTURED RESULTS

A. Captured Results

Fig. S24 shows the hardware captured result of student model for short propagation distance. Fig. S27 shows the hardware captured result of student model for long propagation distance.

B. Comparisons under varying peak brightness

To facilitate direct comparison, this subsection aggregates the hardware-captured results from the previous two subsections into two large figures, allowing side-by-side evaluation with simulations. Fig. S28 presents simulated and captured results of the student model at a short propagation distance ($Z = 2\ mm$). Post-processing is performed using our in-house homography pipeline implemented in Python with OpenCV. Our method preserves fine details, color accuracy, and texture across different s values. Fig. S29 shows the results at a longer propagation distance ($Z = 10\ mm$). Compared with the short-distance case, moderate color deviations appear in the hardware captures. We

Method	Input	Display-scene Parameters	PSNR↑ (dB)		SSIM↑		LPIPS↓	FLIP↓	FVVDP↑	Parameters	Speed
			Mean	Std	Mean	Std	Mean	Mean	Mean		
Our Method (teacher)	RGB-only	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	**29.33**	2.73	0.95	0.03	0.35	**0.10**	8.30	10.74 M	651 ms
		d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.4x	28.78	2.84	0.94	0.03	0.36	0.11	8.23		
		d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.8x	27.65	2.80	0.94	0.04	0.36	0.11	8.20		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	28.16	3.07	0.94	0.03	0.38	0.12	8.18		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.4x	28.15	2.99	0.94	0.03	0.38	0.11	8.19		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.8x	27.86	2.97	0.93	0.04	0.39	0.11	8.18		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x	27.80	3.06	0.93	0.03	0.40	0.12	8.11		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.4x	27.80	2.95	0.93	0.03	0.40	0.12	8.13		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.8x	27.52	2.92	0.92	0.04	0.41	0.13	8.11		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x	26.92	2.71	0.91	0.04	0.46	0.14	7.89		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.4x	26.97	2.66	0.91	0.04	0.47	0.14	7.93		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.8x	26.54	2.56	0.90	0.04	0.48	0.15	7.83		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.0x	28.23	2.87	0.93	0.05	0.39	0.12	8.00		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.4x	27.72	2.94	0.93	0.05	0.40	0.12	7.91		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.8x	26.89	2.91	0.90	0.06	0.40	0.12	7.85		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.0x	27.24	2.91	0.91	0.05	0.42	0.13	7.89		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.4x	27.21	2.94	0.91	0.06	0.42	0.12	7.88		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.8x	26.94	2.91	0.90	0.06	0.42	0.12	7.87		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.0x	27.00	3.01	0.90	0.05	0.44	0.14	7.89		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.4x	26.96	2.92	0.90	0.05	0.44	0.13	7.90		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.8x	26.73	2.88	0.89	0.06	0.44	0.13	7.89		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.0x	26.31	2.69	0.88	0.05	0.49	0.15	7.70		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.4x	26.33	2.65	0.88	0.05	0.49	0.15	7.72		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.8x	25.97	2.60	0.87	0.05	0.50	0.15	7.61		
Our Method (student)	RGB-only	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	28.55	2.88	0.95	0.03	0.35	**0.10**	**8.48**	2.19 M	39 ms
		d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.4x	28.32	2.79	0.94	0.03	0.35	0.10	8.33		
		d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.8x	27.69	2.78	0.94	0.04	0.36	0.11	8.18		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	28.29	3.01	0.94	0.03	0.36	0.11	8.22		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.4x	28.21	2.98	0.94	0.03	0.37	0.11	8.13		
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.8x	28.06	2.97	0.93	0.04	0.37	0.12	8.11		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x	28.28	3.15	0.93	0.03	0.39	0.12	8.15		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.4x	28.03	3.05	0.92	0.03	0.39	0.11	8.09		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.8x	27.93	2.85	0.92	0.04	0.40	0.12	8.03		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x	27.15	2.81	0.91	0.03	0.44	0.15	7.94		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.4x	27.07	2.72	0.91	0.04	0.44	0.15	7.90		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.8x	26.92	2.43	0.90	0.04	0.45	0.16	7.64		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.0x	27.89	2.98	0.93	0.04	0.39	0.12	8.02		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.4x	27.13	3.00	0.92	0.05	0.39	0.12	7.85		
		d_x: 3.74 μm, Z: 2 mm, VD: 8 mm, 1.8x	26.76	2.86	0.91	0.06	0.40	0.13	7.71		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.0x	27.46	2.91	0.91	0.04	0.41	0.13	7.96		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.4x	27.34	2.88	0.90	0.05	0.41	0.13	7.84		
		d_x: 3.74 μm, Z: 4 mm, VD: 8 mm, 1.8x	26.57	2.75	0.89	0.06	0.42	0.14	7.66		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.0x	27.24	3.07	0.90	0.05	0.43	0.13	7.91		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.4x	27.03	2.88	0.90	0.05	0.43	0.13	7.74		
		d_x: 3.74 μm, Z: 7 mm, VD: 8 mm, 1.8x	26.32	2.80	0.88	0.06	0.45	0.14	7.58		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.0x	26.56	2.71	0.89	0.05	0.46	0.15	7.78		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.4x	26.23	2.76	0.88	0.05	0.46	0.16	7.64		
		d_x: 3.74 μm, Z: 10 mm, VD: 8 mm, 1.8x	26.14	2.44	0.88	0.06	0.47	0.16	7.40		
Our Method (RGB-D condition)	RGB-D	d_x: 6.4 μm, Z: *4.88* mm, VD: *6.75* mm, 1.0x[1]	27.73	1.98	0.93	0.02	0.37	0.11	8.64	6.84 M	566 ms
		d_x: *7.19* μm, Z: 2 mm, VD: *4.23* mm, 1.0x[1]	29.83	2.81	0.96	0.02	0.31	0.09	8.79		
		d_x: *4.57* μm, Z: 10 mm, VD: *7.61* mm, 1.0x[1]	26.93	2.19	0.91	0.03	0.42	0.13	8.07		
HoloBeam	RGB-only[2]	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	29.12	2.70	0.93	0.04	0.37	**0.10**	8.37	1.94 M	**27 ms**
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	27.15	2.62	0.90	0.03	0.43	0.11	8.24		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x	25.31	2.24	0.87	0.03	0.47	0.13	8.11		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x	20.62	2.51	0.82	0.03	0.49	0.14	7.99		
Tensor V2	RGB-D	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	29.23	2.26	**0.97**	0.03	**0.34**	**0.10**	8.47	0.04 M	75 ms
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	28.01	2.01	0.95	0.03	0.37	0.10	8.32		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x	25.89	2.09	0.93	0.03	0.39	0.11	8.29		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x	23.04	1.98	0.91	0.03	0.41	0.11	8.12		
modified 3D NH[3]	RGB-D	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	29.01	2.45	0.94	0.03	0.40	**0.10**	8.41	3.87 M	49 ms
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	29.03	2.31	0.92	0.04	0.41	0.11	8.25		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x[4]	28.92	2.18	0.90	0.04	0.43	0.13	8.17		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x[4]	28.98	2.39	0.89	0.04	0.45	0.13	8.05		
Two-stage DepthAny V2	RGB-D	d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, 1.0x	29.21	3.15	0.95	0.03	0.35	**0.10**	8.39	29.2 M	126 ms
		d_x: 3.74 μm, Z: 4 mm, VD: 4 mm, 1.0x	28.65	3.54	0.94	0.04	0.37	0.11	8.14		
		d_x: 3.74 μm, Z: 7 mm, VD: 4 mm, 1.0x[4]	27.92	3.28	0.92	0.04	0.40	0.12	8.07		
		d_x: 3.74 μm, Z: 10 mm, VD: 4 mm, 1.0x[4]	27.37	3.28	0.90	0.05	0.45	0.13	8.00		

Table 6. The extended evaluation table at different holography settings. All the other CGH models in the table, excluding ours, are separately trained with fixed configurations and have no configurability at all. *Parameters* refers to the size of the model. *Speed* refers to the inference time of the model under fp32. Note that *RGB teacher*, *RGB student*, *HoloBeam*, *Tensor V2* and *modified 3D NH* were trained with 800 × 800, 1024 × 1024, 1024 × 1024, 384 × 384 and 1024 × 1024 data, respectively. The test resolution is 1920 × 1080. [1]Novel cases are in italic and *not* included in metrics comparison. [2]We improve the HoloBeam from G-only to RGB-only input. [3]Since NH only supports 2D holograms, we modified NH's HoloNet to generate 3D holograms. [4]A notable disadvantage of the *modified 3D NH* is its sensitivity to resolution under long propagation distances.

attribute this to the larger diffraction cone in long-distance propagation, which increases the required spatial bandwidth and amplifies the impact of limited training resolution, leading to chromatic artifacts. Training at higher resolution can alleviate this effect but significantly increases computational cost. To further investigate this sensitivity, we train an RGB-D conditioned model using full-HD data for $Z = 10\ mm$ and compare it with modified 3D NH, Tensor V2, and Two-stage DepthAny V2.

C. Hardware-captured Result On Other Holographic Displays

Fig. S26 shows the hardware captured result of RGB-D condition on Holoeye Pluto-VIS holographic display (resolution at 1080 $\times$ 1920 and pixel pitch at 8.0 μm). Fig. S25 shows the hardware captured result of RGB-D condition on Holoeye LETO-3 holographic display (resolution at 1920 $\times$ 1080 and pixel pitch at 6.4 μm). Both models are trained under Sec. S17's condition.

17. RGB-D CONDITION MODEL

A. Model Architecture and Training Configuration

To demonstrate that our model structure can condition various pixel pitches, wavelengths, and other display-scene parameters, we derive an RGB-D version of our RGB-only model. Fig. S18 shows the RGB-D version of our model. Similar to the RGB-only model, this model also contains two consecutive Multilayer Perceptron (MLP) layers for conditioning.

We choose the RGB-D approach for extensive parameter variation studies for two key reasons: First, RGB-D input directly provides depth information as prior knowledge, allowing the model to focus on phase-only holography prediction, which greatly reduces data requirements compared to RGB-only models. Second, as the variable set grows larger, the number of permutations increases exponentially, making RGB-D conditioning more computationally tractable.

B. Pair-wise Analysis of Variance (ANOVA)

Pair-wise ANOVA [86] was employed to evaluate the impact of six different condition settings within the RGB-D condition model on image quality metrics. Across PSNR, SSIM, LPIPS, and FLIP, conditions (1)–(4) (pixel pitches of 10.8 μm and 8.0 μm) show no statistically significant differences (p-values ranging from 0.5 to 1.0), indicating that image quality is stable at larger pixel pitches. In contrast, all comparisons involving conditions (5) and (6) (pixel pitch 3.74 μm) yield p-values below 2×10^{-16}, confirming that smaller pixel pitch has a statistically significant impact on reconstruction quality. Tbl. S7 presents the result of a pair-wise ANOVA test conducted between our student model and other models. For PSNR, comparisons with Holobeam and modified 3D NH yield p-values of 0.49 and 0.79 respectively, suggesting statistical equivalence in PSNR stability. For Tensor V2, lower PSNR p-values (0.12) are observed, likely because Tensor V2 was evaluated using its short propagation weight (Z: 0mm, VD: 6mm) rather than a matched configuration, as reimplementing the MIT_CGH_4K dataset [1] for retraining was not feasible. For SSIM and FLIP, p-values below 1×10^{-16} against modified 3D NH and HoloBeam indicate that our student model achieves statistically better perceptual quality (student: SSIM 0.96 ± 0.03, FLIP 0.08 vs. modified 3D NH: 0.94 ± 0.03, 0.11 and HoloBeam: 0.93 ± 0.04, 0.10). Tensor V2 comparisons on SSIM (p=0.02) and FLIP (p=0.01) suggest the two methods are close but not statistically equivalent. For LPIPS, the student model, HoloBeam (p=0.21), and Tensor V2 (p=0.09) are weakly related, with mean LPIPS of 0.35, 0.37, and 0.29 respectively.

	PSNR	SSIM	LPIPS	FLIP
student model & modified 3D NH	0.79	<2e-16	1e-15	1e-9
student model & HoloBeam	0.49	<2e-16	0.21	<2e-16
student model & Tensor V2	0.12	0.02	0.09	0.01

Table 7. The p-value results of pair-wise ANOVA test between student model and various models. The optic setting of the student model, modified 3D NH and HoloBeam is d_x: 3.74 μm, Z: 2 mm, VD: 4 mm, x1.0. The optic setting of Tensor V2 is d_x: 8.0 μm, Z: 0 mm, VD: 6 mm, x1.0.

18. ADDITIONAL ANALYSES

A. Comparison Between HoloBeam and Student Model

Fig. S19 shows the reconstruction comparisons between HoloBeam and student model. When the input data is RGB-only, HoloBeam's results exhibited a deficiency in accurately representing the 3D phase, with incorrect defocus relationships. Conversely, our student model do not have such focusing issues.

B. Impact of Depth Estimation Inaccuracies

Fig. S20 shows that depth estimation errors can cause unintended defocus in regions meant to be sharp at specific depth planes, especially in high-resolution images.

C. Ablation Study: Parameter Embedding

To evaluate the contribution of each component in our parameter embedding layer , we conduct an inference-time ablation study using a trained RGB-D condition model checkpoint. We compare three variants: (a) the full embedding combining sinusoidal scalar encoding and 1D PSF, (b) sinusoidal scalars only, and (c) 1D PSF only. We evaluate on 100 DIV2K test images across four representative configurations spanning short/long propagation distances and two pixel pitches. Results are reported in Tbl. S8.

Embedding Variant	PSNR↑	SSIM↑	LPIPS↓	FLIP↓
d_x=3.74 µm, Z=2 mm, VD=4 mm, ×1.0				
(a) Full (sinusoidal + 1D PSF)	**28.85**	**0.96**	**0.36**	**0.13**
(b) Sinusoidal scalars only	27.52	0.94	0.41	0.15
(c) 1D PSF only	25.13	0.91	0.48	0.19
d_x=3.74 µm, Z=10 mm, VD=4 mm, ×1.0				
(a) Full (sinusoidal + 1D PSF)	**26.41**	**0.94**	**0.40**	**0.15**
(b) Sinusoidal scalars only	25.18	0.92	0.45	0.18
(c) 1D PSF only	23.06	0.88	0.53	0.22
d_x=8.0 µm, Z=2 mm, VD=6 mm, ×1.0				
(a) Full (sinusoidal + 1D PSF)	**31.22**	**0.97**	**0.30**	**0.07**
(b) Sinusoidal scalars only	30.05	0.96	0.34	0.09
(c) 1D PSF only	27.84	0.93	0.42	0.12
d_x=8.0 µm, Z=10 mm, VD=6 mm, ×1.0				
(a) Full (sinusoidal + 1D PSF)	**28.76**	**0.95**	**0.35**	**0.10**
(b) Sinusoidal scalars only	27.61	0.93	0.39	0.12
(c) 1D PSF only	25.49	0.90	0.47	0.16

Table 8. Ablation of the parameter embedding layer.

The full embedding consistently achieves the best performance across all four configurations. Removing the 1D PSF branch (variant b) causes an average PSNR drop of 1.3 dB. Removing the sinusoidal scalars (variant c) leads to a larger average drop of 3.5 dB, as the 1D PSF alone cannot disambiguate configurations that share similar PSF shapes but differ in wavelength or volume depth. These results confirm that both branches provide complementary information. The sinusoidal scalars supply exact parameter values while the 1D PSF injects physics-aware spatial structure.

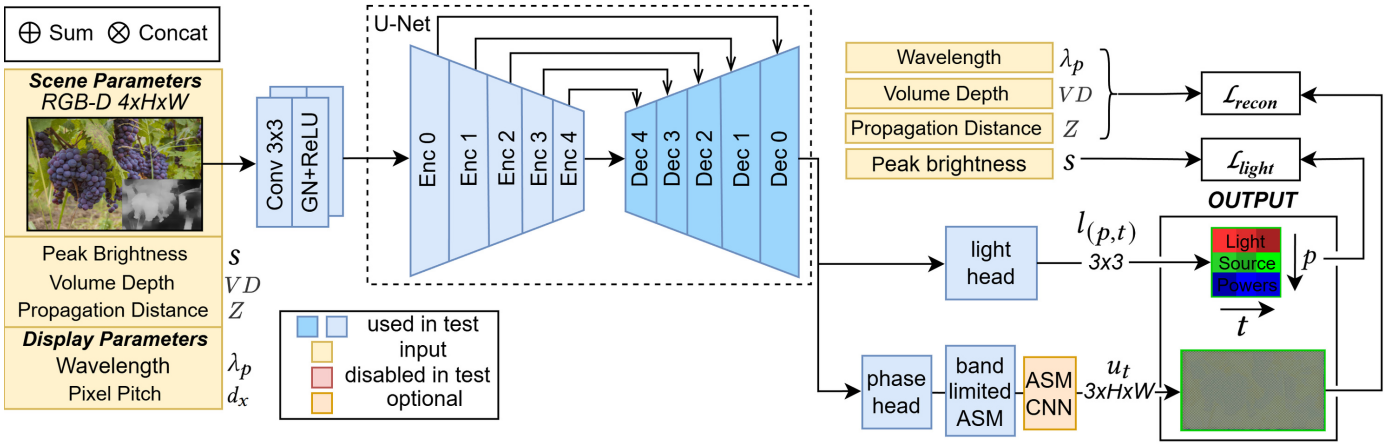


Fig. 18. The overview of the RGB-D condition model. (RGB input: [33])

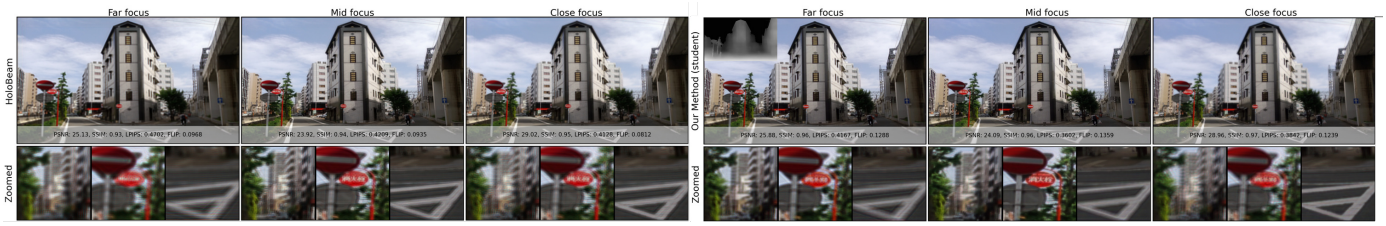


Fig. 19. The simulated reconstructions comparison between HoloBeam and student model. The volume depth of the results is 4mm and the propagation distance is 2mm (Source Image: [87]).

Fig. 20. Depth estimation errors causing incorrect defocus in high-resolution images. The student model predicts the top of the price board at the mid-focus plane, leading to erroneous reconstructions in the mid- and near-focus planes. Blue indicates the mid-focus region and green the near-focus region. (Source image: [88]).

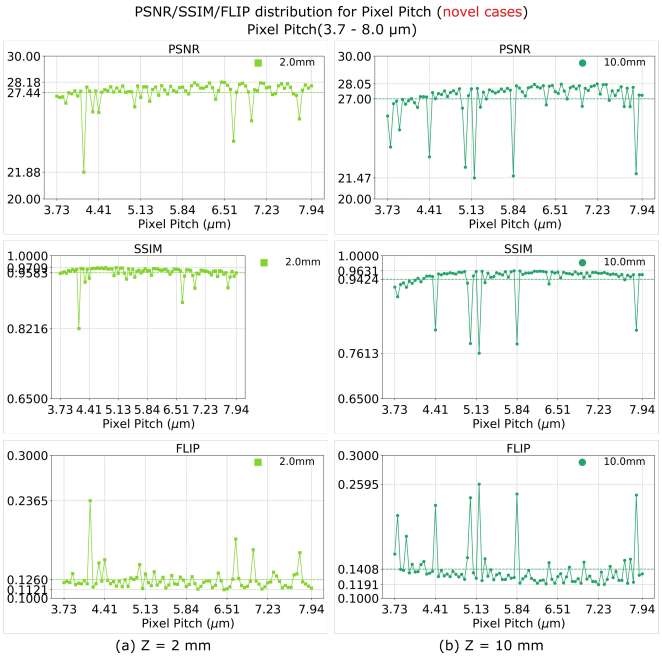
19. VARYING DISPLAY-SCENE PARAMETERS

A. Generalizing Novel Cases Outside Of Training (Pixel Pitch)

Our model can generalize novel pixel pitch values when the distance between the training conditions is small. Empirically, we found that the model has the best performance when the step size between cases is around 0.05 μm. We trained the RGB-D condition model with the following conditions:

$$\begin{aligned} \lambda_p &\subseteq \{(640, 515, 470)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{4.0, 8.0\}\ mm, \\ Z &\subseteq \{2.0, 10.0\}\ mm, d_x \subseteq \{3.7 - 8.0\}\ \mu m. \end{aligned} \tag{20}$$

In this experiment, we include the pixel pitch covering a span of 4.3 μm range, 2 propagation distance, and a volume depth as large as 4mm. The pixel pitch conditions have a 0.05



(a) Z = 2 mm (b) Z = 10 mm

Fig. 21. The PSNR, SSIM, and FLIP distribution of our RGB-D condition model when conditioned on a 4.3 μm total pixel pitch range across two propagation distances.

μm step size, which results in 54 conditions in total. The entire permutation set contains 344 conditions in total and the model is trained at the resolution of 896 $\times$ 896.

To conduct a comprehensive evaluation of the model, we employ an uniform sampling approach across the entire range of pixel pitch. For each pixel pitch interval (e.g. 4-5 μm), we generate a set of 20 novel pixel pitches outside of the training set. These randomly selected pixel pitches are uniformly distributed within the interval, cases are approximately 0.05 μm between each other. We use the following novel cases as test cases:

$$\begin{aligned} \lambda_p &\subseteq \{(640, 515, 470)\}\ nm, \\ s \subseteq \{1.0\}, VD &\subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{2.0, 10.0\}\ mm, \\ d_x &\subseteq \{\text{randomized pixel pitch distribution}\}\ \mu m. \end{aligned} \tag{21}$$

The test permutation set contains 344 novel cases at a resolution of 1280 $\times$ 1280. Due to the large number of conditions, tabular reporting is impractical; instead, we present the results using figures. Fig. S21 reports PSNR, SSIM, and FLIP distributions, where each data point is computed from the same 100 images from the DIV2K dataset [64]. Overall, the model maintains stable image quality with low standard deviation across randomly sampled pixel pitch conditions over a continuous range. About 10% of cases show noticeable quality variations, likely due to the limited model capacity when generalizing d_x across a wide continuous range (4.3 μm), which is more challenging than other display–scene parameters and makes adapting to hundreds of conditions difficult.

B. Generalizing Novel Cases Outside Of Training (Propagation Distance)

Similar to pixel pitch, our model can generalize novel propagation distance values when the distance between the training conditions is small. Empirically, we found that the model has the best performance when the step size between cases is around 0.005 mm. We trained the RGB-D condition model with the following conditions:

$$\begin{aligned} \lambda_p &\subseteq \{(640, 515, 470)\}\ nm, \\ s \subseteq \{1.0\}, VD &\subseteq \{4.0, 8.0\}\ mm, \\ Z \subseteq \{2.0 - 3.0; 4.0 - 5.0; &6.0 - 7.0; \\ &8.0 - 9.0; 10.0 - 11.0\}\ mm, \\ d_x &\subseteq \{3.74, 6.4, 8.0\}\ \mu m. \end{aligned} \tag{22}$$

In this experiment, we include the propagation distance covering a span of 5mm range, 3 common pixel pitches, and a volume depth as large as 4mm. The training propagation distance conditions are 0.005 mm between each others, which results in 1000 conditions in total. The entire permutation set contains 6000 conditions in total and the model is trained at the resolution of 512 $\times$ 512. We choose a discrete, rather than a continuous, 5mm range because we want to maximize the Z range coverage while avoiding excessive computational demands. The conditioning of a continuous 5mm range of Z will also work under our training setting. To conduct a comprehensive evaluation of the model, we employ an uniform sampling approach across the entire range of propagation distances. For each propagation distance interval (e.g. 2 - 3 mm), we generate a set of 100 novel propagation distances outside of the training set. These randomly selected distances are uniformly distributed within the interval, cases are approximately 0.01mm between each other. We use the following novel cases as test cases:

$$\begin{aligned} \lambda_p &\subseteq \{(640, 515, 470)\}\ nm, \\ s \subseteq \{1.0\}, VD &\subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{\text{randomized distance distribution}\}\ mm, \\ d_x &\subseteq \{3.74, 6.4, 8.0\}\ \mu m. \end{aligned} \tag{23}$$

The test permutation set contains 3000 novel cases at a resolution of 1024 $\times$ 1024. Due to the large number of conditions, tabular evaluation is impractical; therefore, results are presented in figure form. Fig. S23 shows the PSNR distribution, while SSIM and FLIP follow the same trend and are omitted for brevity. Each data point is computed from the same 100 images from the DIV2K dataset [64]. Overall, the model maintains stable image quality with low standard deviation across randomly generated propagation distances over a continuous range. About 20% of cases show noticeable quality variations, likely due to the limited model capacity when generalizing continuous Z over a wide range (5 mm), which is more challenging than other display–scene parameters and makes adapting to thousands of conditions difficult.

C. Generalizing Novel Cases Outside Of Training (Wavelength)

As demonstrated in Sec. S17, our model structure effectively adapts to a wide range of wavelengths. In this section, we focus on showcasing the model's capability for continuous wavelength generalization under the RGB-D condition. Our model can generalize novel wavelength values continously. Empirically, we found that the model has the best performance when the step size between cases is around 5 nm. We trained the RGB-D condition model with the following conditions:

$$\begin{aligned} \lambda_p &\subseteq \{(625-680, 510-565, 425-480)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{4.0, 8.0\}\ mm, \\ Z &\subseteq \{2.0, 10.0\}\ mm, d_x \subseteq \{3.74, 8.0\}\ \mu m. \end{aligned} \tag{24}$$

In this experiment, we consider wavelengths spanning a 55 nm range, two propagation distances, two common pixel pitches, and a volume depth up to 4 mm. Wavelength conditions are sampled at 5 nm intervals, and the model is trained at a resolution of 640×640. Given the large number of possible wavelength combinations (a 55 nm range yields $55^3 = 166375$ combinations), we evaluate the model on twenty randomly sampled novel cases at a test resolution of 1024×1024:

Condition Set (1)

$$\begin{aligned} \lambda_p &\subseteq \{(641, 546, 478), (668, 547, 461), \\ &(676, 563, 441), (632, 538, 473), (659, 519, 432)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{2.0\}\ mm, d_x \subseteq \{3.74\}\ \mu m. \end{aligned} \tag{25}$$

Condition Set (2)

$$\begin{aligned} \lambda_p &\subseteq \{(661, 558, 433), (628, 554, 480), \\ &(631, 525, 474), (653, 562, 471), (679, 521, 462)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{10.0\}\ mm, d_x \subseteq \{3.74\}\ \mu m. \end{aligned} \tag{26}$$

Condition Set (3)

$$\begin{aligned} \lambda_p &\subseteq \{(668, 518, 426), (628, 510, 444), \\ &(678, 557, 467), (639, 541, 437), (655, 531, 434)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{2.0\}\ mm, d_x \subseteq \{8.0\}\ \mu m. \end{aligned} \tag{27}$$

Condition Set (4)

$$\begin{aligned} \lambda_p &\subseteq \{(659, 523, 454), (663, 562, 466), \\ &(677, 515, 443), (628, 542, 429), (642, 557, 457)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{\text{randomized between 4.0 - 8.0}\}\ mm, \\ Z &\subseteq \{10.0\}\ mm, d_x \subseteq \{8.0\}\ \mu m. \end{aligned} \tag{28}$$

Tbl. S9 shows our model's performance across twenty test conditions with different randomized RGB wavelengths under various combinations of propagation distance and pixel pitch. The model can take arbitrary wavelengths as the input and maintain consistent image quality over different propagation distances and pixel pitches. Each data in the table is contributed by the same 100 images from the DIV2K dataset [64].

D. Effect of Conditioning Range on Quality Stability

In the main paper and Sec. S19, we observe quality fluctuations when the model conditions on propagation distances across a wide 5 mm total range (five disjoint 1 mm intervals). To investigate whether reducing the total conditioning range mitigates these fluctuations, we train an RGB-D condition model on a single continuous 1 mm interval from 6–7 mm with all other parameters fixed:

$$\begin{aligned} \lambda_p &\subseteq \{(644, 519, 468)\}\ nm, \\ s &\subseteq \{1.0\}, VD \subseteq \{4.0, 8.0\}\ mm, \\ Z &\subseteq \{6.0-7.0\}\ mm, d_x \subseteq \{3.74\}\ \mu m. \end{aligned} \tag{29}$$

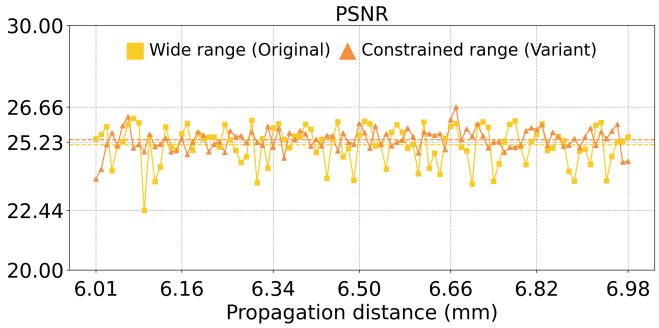


Fig. 22. PSNR distribution over 100 novel propagation distances at $d_x = 3.74\,\mu$m. Each point is the mean PSNR over 100 DIV2K test images at a randomly sampled (VD).

The training propagation distance conditions are sampled at 0.005 mm intervals within the 6–7 mm range, yielding 200 Z conditions. Combined with two VD values, the total training permutation set contains 400 conditions. The model is trained at a resolution of 512×512. For evaluation, we generate 100 novel propagation distances randomly sampled within 6–7 mm (approximately 0.01 mm apart), each paired with a randomly sampled VD in $[4.0, 8.0]$ mm. The test resolution is 1024×1024, and each data point is the mean PSNR over the same 100 DIV2K images [64]. Fig. S22 compares the PSNR distribution of the constrained-range model (1 mm, 6–7 mm) against the corresponding 6–7 mm interval from the wide-range model (5 mm total, five disjoint 1 mm intervals). The constrained-range model achieves a mean PSNR of 25.33 dB with a standard deviation of 0.44 dB, compared to 25.13 dB mean and 0.78 dB standard deviation for the same Z interval extracted from the wide-range model. Moreover, the minimum per-condition PSNR rises from 22.44 dB to 23.72 dB, indicating that reducing the total conditioning range can suppress the quality drops observed in the wide-range setting.

Method	Input	Display-scene Parameters	PSNR↑ (dB)		SSIM↑		LPIPS↓	FLIP↓	FVVDP↑
			Mean	Std	Mean	Std	Mean	Mean	Mean
Our Method (RGB-D condition)	RGB-D	λ: (*641*,*546*,*478*) nm, Z: 10 mm, VD: *4.2* mm, d_x: 3.74 μm, x1.0	25.10	2.52	0.90	0.03	0.43	0.19	7.41
		λ: (*668*, *547*, *461*) nm, Z: 10 mm, VD: *5.3* mm, d_x: 3.74 μm, x1.0	24.90	2.51	0.89	0.03	0.44	0.19	7.36
		λ: (675, *563*, *441*) nm, Z: 10 mm, VD: *7.1* mm, d_x: 3.74 μm, x1.0	24.58	2.47	0.87	0.04	0.46	0.20	7.24
		λ: (*632*, *538*, *473*) nm, Z: 10 mm, VD: *5.6* mm, d_x: 3.74 μm, x1.0	24.94	2.50	0.89	0.03	0.44	0.19	7.37
		λ: (*659*, *519*, *432*) nm, Z: 10 mm, VD: *6.4* mm, d_x: 3.74 μm, x1.0	24.76	2.51	0.88	0.04	0.45	0.19	7.35
		λ: (*661*, *558*, *433*) nm, Z: 2 mm, VD: *6.9* mm, d_x: 3.74 μm, x1.0	26.11	2.93	0.92	0.04	0.37	0.14	7.87
		λ: (*628*, *554*, 480) nm, Z: 2 mm, VD: *4.5* mm, d_x: 3.74 μm, x1.0	27.06	2.99	0.94	0.03	0.34	0.12	8.17
		λ: (*631*, *521*, 475) nm, Z: 2 mm, VD: *5.8* mm, d_x: 3.74 μm, x1.0	26.53	2.94	0.93	0.03	0.35	0.13	8.04
		λ: (*653*, *562*, *471*) nm, Z: 2 mm, VD: *6.1* mm, d_x: 3.74 μm, x1.0	26.35	2.93	0.93	0.03	0.36	0.13	7.96
		λ: (*679*, *521*, *462*) nm, Z: 2 mm, VD: *7.8* mm, d_x: 3.74 μm, x1.0	25.76	2.94	0.91	0.05	0.39	0.14	7.80
		λ: (*668*, *518*, *426*) nm, Z: 10 mm, VD: *4.1* mm, d_x: 8.0 μm, x1.0	28.15	3.12	0.95	0.03	0.30	0.12	8.19
		λ: (*628*, 510, *444*) nm, Z: 10 mm, VD: *5.7* mm, d_x: 8.0 μm, x1.0	28.27	3.14	0.95	0.02	0.29	0.12	8.23
		λ: (*678*, *557*, *467*) nm, Z: 10 mm, VD: *4.7* mm, d_x: 8.0 μm, x1.0	28.22	3.14	0.95	0.02	0.30	0.12	8.21
		λ: (*639*, *541*, *437*) nm, Z: 10 mm, VD: *6.5* mm, d_x: 8.0 μm, x1.0	28.32	3.16	0.95	0.02	0.30	0.12	8.25
		λ: (655, *531*, *434*) nm, Z: 10 mm, VD: *7.3* mm, d_x: 8.0 μm, x1.0	28.35	3.18	0.95	0.02	0.30	0.12	8.27
		λ: (*659*, *523*, *454*) nm, Z: 2 mm, VD: *4.9* mm, d_x: 8.0 μm, x1.0	28.45	3.11	0.96	0.02	0.26	0.11	8.40
		λ: (*663*, *562*, *466*) nm, Z: 2 mm, VD: *5.6* mm, d_x: 8.0 μm, x1.0	28.40	3.13	0.96	0.02	0.26	0.11	8.41
		λ: (*677*, 515, *443*) nm, Z: 2 mm, VD: *6.9* mm, d_x: 8.0 μm, x1.0	28.29	3.18	0.96	0.02	0.27	0.11	8.40
		λ: (*628*, *542*, *429*) nm, Z: 2 mm, VD: *7.1* mm, d_x: 8.0 μm, x1.0	28.29	3.19	0.96	0.02	0.27	0.11	8.40
		λ: (*642*, *557*, *457*) nm, Z: 2 mm, VD: *4.6* mm, d_x: 8.0 μm, x1.0	28.47	3.11	0.96	0.02	0.26	0.11	8.40

Table 9. Evaluation of RGB-D condition model with random wavelengths and VD values at different pixel pitches, and propagation distances settings. The test resolution is at 1024 × 1024. The novel cases' (outside of training set) metrics are marked in red.

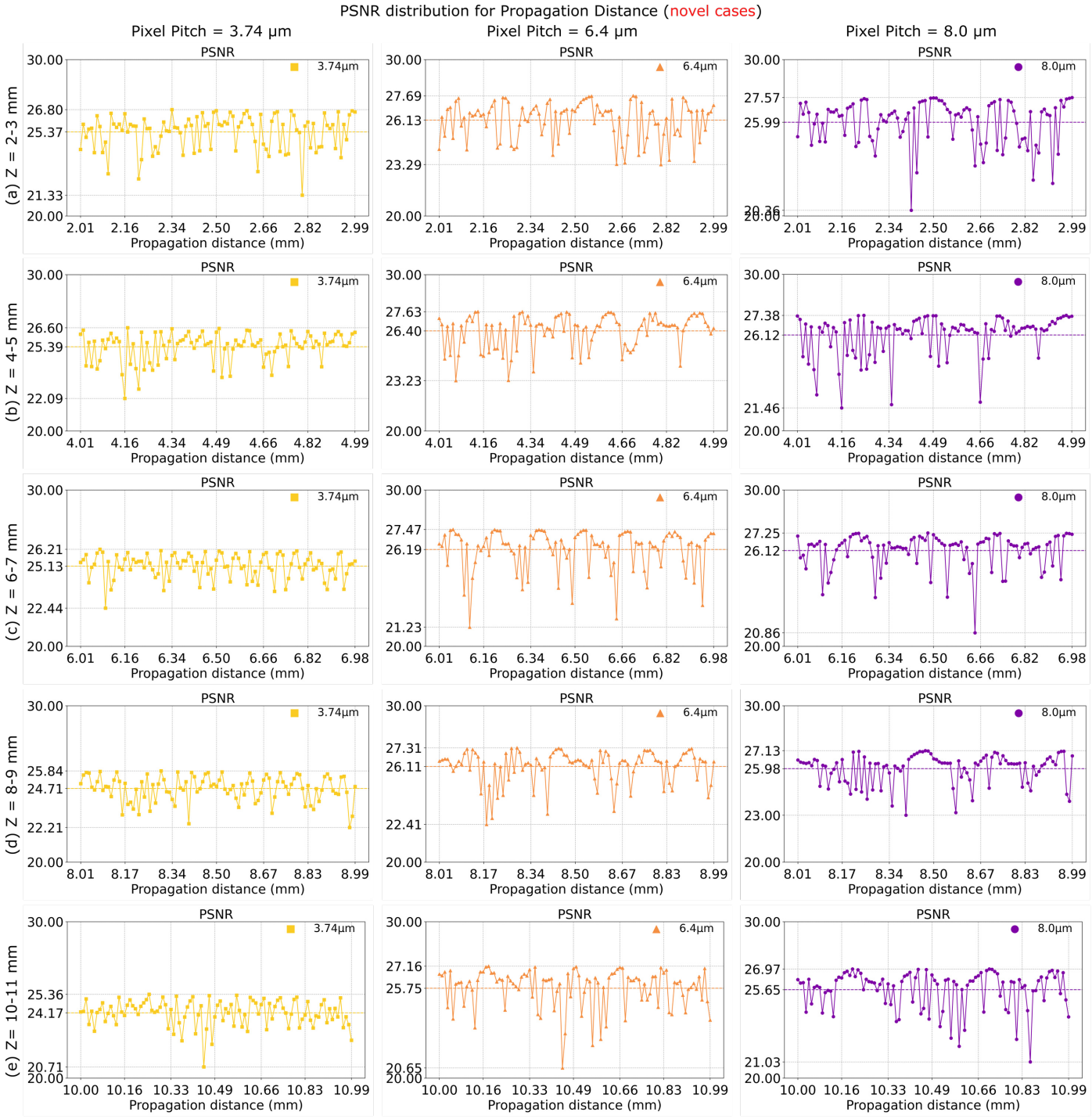


Fig. 23. PSNR distribution of our RGB-D condition model across novel propagation distances for three pixel pitches (3.74, 6.4, 8.0 μm). Each panel covers a continuous 1 mm Z interval; each data point is the mean PSNR over 100 DIV2K test images at a randomly sampled novel (Z, VD) pair. The model maintains consistent quality (average PSNR $\approx$ 26 dB, average std $\approx$ 1.1 dB) with isolated drops in roughly 20% of cases, which we attribute to limited model capacity over the wide conditioning range. SSIM and FLIP distributions follow the same trend and are omitted for brevity.

Fig. 24. The hardware-captured short propagation result of student model when peak brightness set to 1.0, 1.4 and 1.8. From left to right: (Image Source of the first example: [Bernard Spragg 2009]) (Image Source of the second example: [LogicalRailfan 2023]) (Image Source of the third example: [Pilettes 2011])

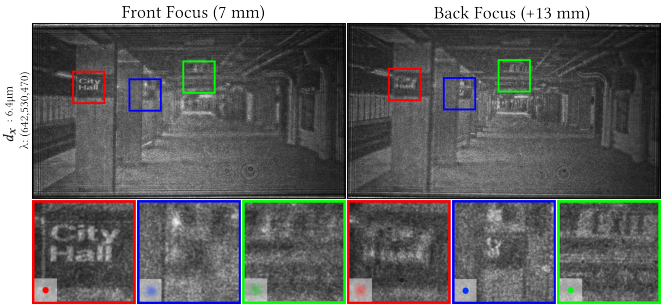


Fig. 25. The hardware-captured result of RGB-D condition model on Holoeye LETO-3 (pixel pitch = 6.4 μm) with volume depth 6 mm. Due to maintenance issue, the result was captured using a green laser only. (Source Image: [65]).

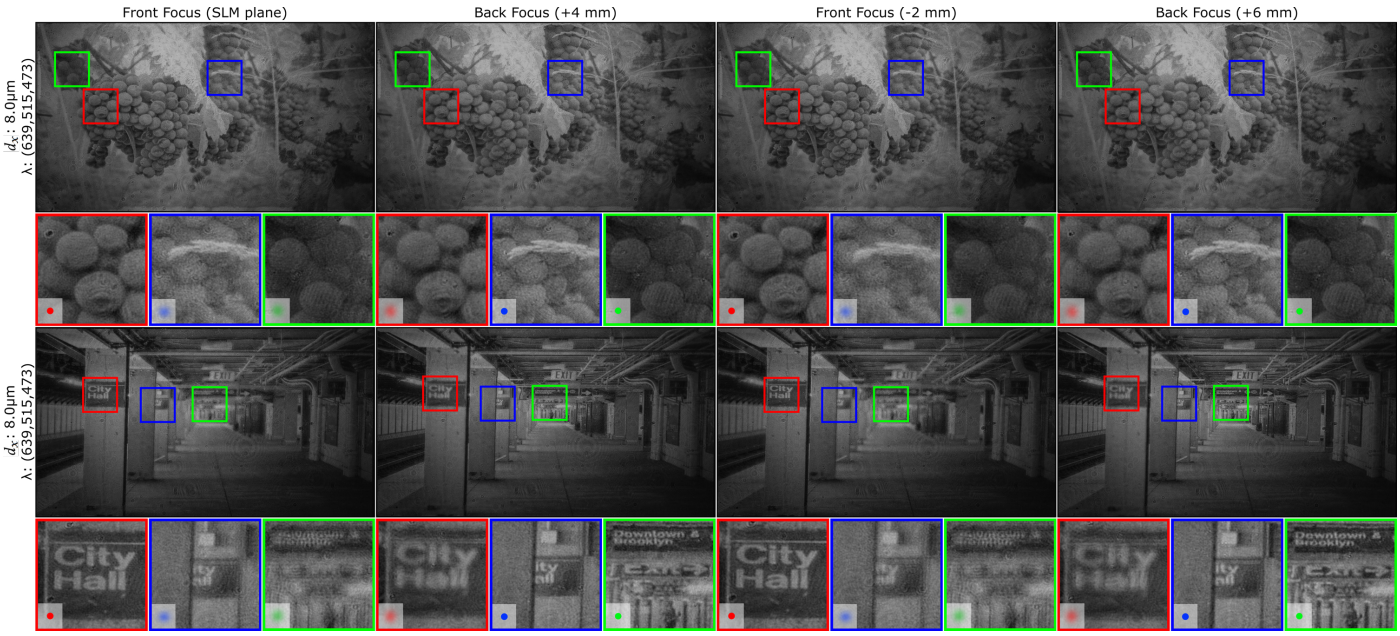


Fig. 26. The hardware-captured result of RGB-D condition model on Holoeye Pluto-VIS (pixel pitch = 8.0 μm) when volume depth set to 4.0 and 8.0 mm. Due to maintenance issue, the result was captured using a red laser only. From top to bottom: (Image Source of the top example: [33]) (Image Source of the bottom example: [87])

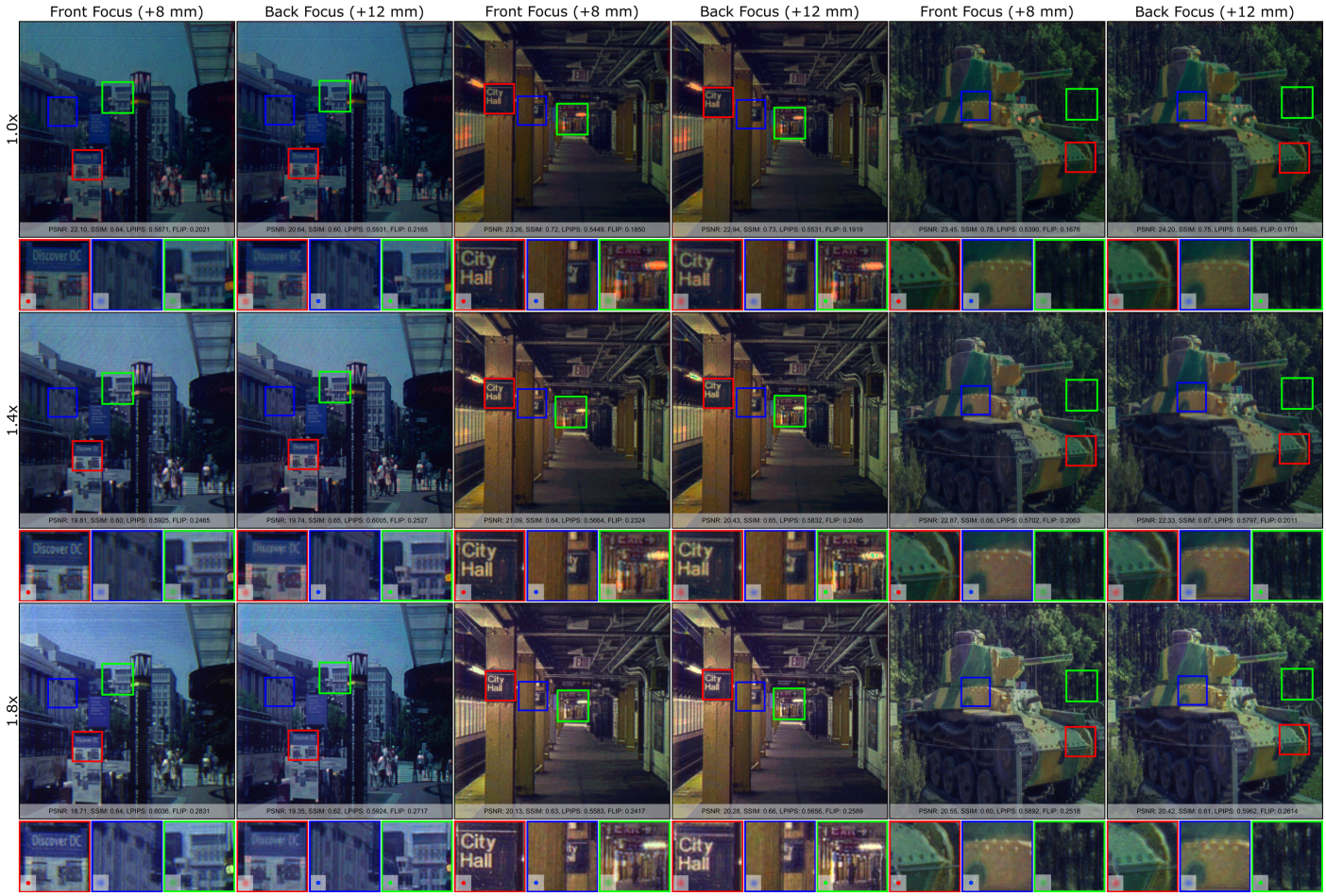


Fig. 27. The hardware-captured long propagation result of student model when peak brightness set to 1.0, 1.4 and 1.8. From left to right: (Image Source of the first example: [Steve Tatum 2010]) (Image Source of the second example: [LogicalRailfan 2023]) (Image Source of the third example: [Mike1979 Russia 2014])

Fig. 28. The simulated and captured short Z distance reconstructions comparison of student model (RGB-only) when peak brightnesses are 1.0, 1.4, and 1.8. The volume depth of the results is 4 *mm* and the propagation distance is 2 *mm*. The resolution of the tested hologram is 2816 $\times$ 2816. All the photographs are captured at 16.6 *ms* exposure time (Source Image: [65]).

Fig. 29. The simulated and captured long Z distance reconstructions comparison of student model (RGB-only) when peak brightnesses are 1.0, 1.4, and 1.8. The volume depth of the results is 4 *mm* and the propagation distance is 10 *mm* The resolution of the tested hologram is 2816 $\times$ 2816. All the photographs are captured at 16.6 *ms* exposure time (Source Image: [65]).