Supplemental Material for HoloPathTracer: Fast and Accurate Wave Path Tracing for Holography

WENBIN ZHOU*, The University of Hong Kong, Hong Kong SAR
XIANGYU MENG*, The University of Hong Kong, Hong Kong SAR
JIANKAI XING, The University of Hong Kong, Hong Kong SAR and Tsinghua University, China
XIN LIU, The University of Hong Kong, Hong Kong SAR
SUYEON CHOI, Stanford University, USA and Seoul National University, Republic of Korea
YIFAN PENG, The University of Hong Kong, Hong Kong SAR

This document provides additional mathematical derivations, algorithm details, prototype details, and results complementary to the main manuscript. We have also provided a *supplementary video* to illustrate the varying depth- and view-dependent effects supported by different CGH frameworks. We will release the data and all necessary code to facilitate the reproducibility of the presented results.

ACM Reference Format:
Wenbin Zhou, Xiangyu Meng, Jiankai Xing, Xin Liu, Suyeon Choi, and Yifan Peng. 2026. Supplemental Material for HoloPathTracer: Fast and Accurate Wave Path Tracing for Holography. *ACM Trans. Graph.* 45, 4, Article 39 (July 2026), 10 pages. https://doi.org/10.1145/3811351

S1 Mathematics and Algorithms Elaboration

S1.1 Near-Field Wave Propagation in Free Space

The wave propagation between the SLM plane and the observation plane is modeled using the established angular spectrum method (ASM) [Goodman 2005; Matsushima and Shimobaba 2009], which can be described as:

$$E(x,y,z) = \mathscr{F}^{-1}\left\{\mathscr{F}\left[u(x_0,y_0)\, e^{i\varphi(x_0,y_0)}\right] H\left(f_x,f_y;z\right)\right\}, \tag{S1}$$

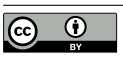
where E and u denote the optical (wave) fields incident onto the observation plane and the SLM, respectively. $\mathscr{F}\left[u(x_0,y_0)\, e^{i\varphi(x_0,y_0)}\right]$ is the so-called angular spectrum of $u(x_0,y_0)\, e^{i\varphi(x_0,y_0)}$. Notably, u is often assumed to be a plane wave. φ denotes the phase modulation, i.e., the hologram, induced by the SLM. $\mathscr{F}\{\cdot\}$ and $\mathscr{F}^{-1}\{\cdot\}$ are the Fourier transform operator and its inverse, respectively. H is the transfer function in the frequency domain, defined as:

$$H\left(f_x,f_y;z\right) = e^{i2\pi z\sqrt{\frac{1}{\lambda^2}-f_x^2-f_y^2}}, \tag{S2}$$

where λ indicates the concerned wavelength.

*denotes equal contribution.

Authors' Contact Information: Wenbin Zhou, The University of Hong Kong, Hong Kong SAR, zhouwb@connect.hku.hk; Xiangyu Meng, The University of Hong Kong, Hong Kong SAR, mengxy22@connect.hku.hk; Jiankai Xing, The University of Hong Kong, Hong Kong SAR and Tsinghua University, China, xjk21@mails.tsinghua.edu.cn; Xin Liu, The University of Hong Kong, Hong Kong SAR, liuxin24@hku.hk; Suyeon Choi, Stanford University, USA and Seoul National University, Republic of Korea, suyeon@stanford.edu; Yifan Peng, The University of Hong Kong, Hong Kong SAR, evanpeng@hku.hk.



ACM 1557-7368/2026/7-ART39
https://doi.org/10.1145/3811351

S1.2 Wave Path Tracing

In Fig. 8 of the main manuscript, we illustrate ray–object interactions in world space. Rays originating from the recording plane (camera) scatter at object surfaces before reaching the light source. Upon striking a facet, the scattering type is selected probabilistically among the available lobes (e.g., specular reflection, ideal refraction, diffuse reflection, glossy reflection). The outgoing ray direction is then sampled from the probability distribution prescribed by the selected lobe, as described at image point P_2.

When a ray strikes a facet and a non-δ lobe is selected for scattering (as at image point P_1), multiple outgoing rays are sampled from the probability distribution of that lobe. These rays provide Monte Carlo estimates of the incoherent radiance. Their radiances are first averaged to estimate the outgoing intensity at P_1, and the corresponding linear amplitude is then obtained by taking the square root. After appending a random phase to model incoherence, this result is treated as the complex amplitude propagated back along the incident ray to P_1.

When a ray strikes a facet and a δ-lobe is selected, no directional scattering is sampled: the ray undergoes ideal reflection or refraction. The phase distribution of the light is preserved, consistent with Huygens' principle. Before and after traversing a δ-lobe, the coherent plane-wave bundle associated with the ray is preserved and propagates back toward the camera (as illustrated in the lower-left of Fig. 8). In contrast, once a ray undergoes an incoherent non-δ scattering event, it no longer carries a coherent plane-wave bundle but instead transports incoherent radiance. This behavior can be illustrated by the two orange ray branches: one undergoes non-δ scattering at P_2, while the other experiences δ specular reflection at P_2, then reaches P_3, where it undergoes non-δ scattering.

The **pseudocode** in Algorithms S1 and S2 provides a detailed description of the proposed wave path tracing pipeline, including ray sampling from the hologram plane, recursive scene traversal, and the treatment of coherent δ-lobe transport versus incoherent non-δ scattering. For brevity, we have omitted Russian roulette and multi-importance sampling, which are commonly used in path tracing to reduce variance and accelerate convergence.

S1.3 Bias Analysis of the Estimator

For the estimator itself, scattering is unbiased between incoherent multi-frame superposition and BSDF-based rendering. Consider an elemental scattering patch dA and one time-multiplexed frame t. Its

Algorithm S 1: Wave Path Tracing

Data: 3D scene $\mathcal{S}$, Samples Per Pixel N_{spp}, Samples Per Non-δ lobe $N_{\text{scattering}}$, maximum ray depth $\text{depth}_{\max}$, coordinate transformation $\mathbf{T}$, wavelength λ, feature size d_x
Result: complex hologram $U(x, y)$

foreach $O_{\text{SLM}}(x, y, 0)$ *on hologram plane* **do**
  $U(x, y) \leftarrow 0$;
  for $i \leftarrow 1$ **to** N_{spp} **do**
    $\mathbf{u} \leftarrow \text{uniformSample}(\arcsin \frac{\lambda}{2d_x})$;
    // Uniformly sample $\mathbf{u}$ on the spherical cap where $\arccos(\mathbf{u} \cdot \mathbf{n}) < \arcsin \frac{\lambda}{2d_x}$
    $U \leftarrow O_{\text{SLM}} + \mathbf{u}$;
    $V \leftarrow \mathbf{T}^{-1}U, O_{\text{world}} \leftarrow \mathbf{T}^{-1}O_{\text{SLM}}$;
    $\mathbf{v} \leftarrow \text{normalize}(\overrightarrow{O_{\text{world}}V})$;
    $L, d_{\text{OPL}} \leftarrow \texttt{trace}(O_{\text{world}}, \mathbf{v}, \text{depth} = 0, \delta = \text{True})$;
    $A = \sqrt{L}$;
    $P' \leftarrow O_{\text{world}} + d_{\text{OPL}}\mathbf{v}$;
    $Q \leftarrow \mathbf{T}P'$;
    $\varphi_r \leftarrow \text{GaussianRandomField}(x_Q, y_Q)$;
    $U(x, y) \leftarrow U(x, y) + Ae^{i(kd_{\text{OPL}})+\varphi_r}$;
    // k is wavenumber, yielding $k = 2\pi/\lambda$
  $U(x, y) \leftarrow U(x, y)/N_{\text{spp}}$;
return $U(x, y)$

Algorithm S 2: Trace function

Function $\texttt{trace}(s, \mathbf{v}, \text{depth}, \delta)$**:**
  // This function traces the ray starting from s in direction $\mathbf{v}$,
  // and returns the intensity L and accumulated OPL d_{OPL} carried by this ray
  // Depth denotes the ray tracing recursion level,
  // and δ = True indicates that the ray has yet to encounter a non-δ lobe
  $t \leftarrow \text{intersect}(s, \mathbf{v}; \mathcal{S})$;
  // Compute the intersection t between the ray and the scene $\mathcal{S}$
  $d_{\text{OPL}} \leftarrow n|t - s|$;
  // n is the refractive index
  $L \leftarrow \text{emittedRadiance}(t)$;
  // emissive term, note that $L = A^2$
  if $\text{depth} > \text{depth}_{\max}$ **then**
    // Reached the maximum bounce limit
    return L, d_{OPL} ;
  $\text{Lobe} \leftarrow \text{sampleLobe}(t)$;
  if δ = True **then**
    if Lobe *is* δ **then**
      $\mathbf{r} \leftarrow \text{deltaScattering}(t, \mathbf{v}; \text{Lobe})$;
      $L', d'_{\text{OPL}} \leftarrow \texttt{trace}(t, \mathbf{r}, \text{depth} + 1, \delta = \text{True})$;
      $d_{\text{OPL}} \leftarrow d'_{\text{OPL}} + d_{\text{OPL}}$;
      $L \leftarrow L'$;
      // emissive term is discarded
    else
      $L_{\text{sum}} \leftarrow 0$;
      for $j \leftarrow 1$ **to** $N_{\text{scattering}}$ **do**
        $\mathbf{r} \leftarrow \text{sampleScattering}(t, \mathbf{v}; \text{Lobe})$;
        $L', d'_{\text{OPL}} \leftarrow \texttt{trace}(t, \mathbf{r}, \text{depth} + 1, \delta = \text{False})$;
        // d_{OPL} is discarded
        $L_{\text{sum}} \leftarrow L_{\text{sum}} + L'$;
      $L \leftarrow L_{\text{sum}}/N_{\text{scattering}} + L$;
  else
    if Lobe *is* δ **then**
      $\mathbf{r} \leftarrow \text{deltaScattering}(t, \mathbf{v}; \text{Lobe})$;
    else
      $\mathbf{r} \leftarrow \text{sampleScattering}(t, \mathbf{v}; \text{Lobe})$;
    $L', d'_{\text{OPL}} \leftarrow \texttt{trace}(t, \mathbf{r}, \text{depth} + 1, \delta = \text{False})$;
    $L \leftarrow L'$;
  return L, d_{OPL};

angular spectrum can be written as

$$U_{dA,t}(f_x, f_y) = \mathscr{F}\left\{ \iint I(\omega_i)\,\text{BSDF}(\omega_i, \omega_o)\, e^{i\phi_{\text{rand}}(\omega_o,t)}\, e^{i2\pi(xf_x+yf_y)}\, d\omega_o \right\}$$
$$= I(\omega_i)\,\text{BSDF}\big(\omega_i, \omega_o(f_x, f_y)\big)\, e^{i\phi_{\text{rand}}(\omega_o(f_x,f_y),t)}, \tag{S3}$$

where $I(\omega_i)$ denotes the incident radiance and $\phi_{\text{rand}}(\omega_o, t)$ is the random phase assigned in frame t. Since the random phase only modulates the phase term while preserving the angular amplitude, the expectation over frames satisfies

$$\mathbb{E}_t\left[\left|U_{dA,t}(f_x, f_y)\right|\right] = I(\omega_i)\,\text{BSDF}\big(\omega_i, \omega_o(f_x, f_y)\big)\,. \tag{S4}$$

Hence, under incoherent time integration, the multi-frame hologram reconstruction recovers the target angular distribution, or equivalently the target light-field intensity distribution, in expectation.

S1.3.1 δ/non-δ split. The remaining deviations observed in reconstruction results come from certain reasonable approximations existing in current numerical implementations. The most important one arises from the δ/non-δ split used in the current renderer. Our objective is not to reproduce the exact interference pattern of every coherent ray emitted in a fully coherent optical setup, but to reconstruct the physically correct image/object points produced by a scene under incoherent illumination. For this reason, the implementation classifies each sampled BSDF lobe according to whether it preserves optical path length and imaging ability: δ lobes preserve both and are traced coherently, whereas non-δ lobes are treated as incoherent scattering events whose subsequent transport is represented only through radiance. This approximation is accurate for fully diffuse or ideal specular interactions, but low-roughness glossy materials lie between these two regimes and may still retain limited imaging ability. As a result, the current binary treatment can introduce a bias for such materials.

Two refinements are possible. One possible refinement is to replace the hard binary decision with a roughness-dependent probabilistic weighting scheme, in which each glossy lobe is sampled as either a coherent plane wave bundle or an incoherent radiance ray according to probabilities determined by its roughness. In this

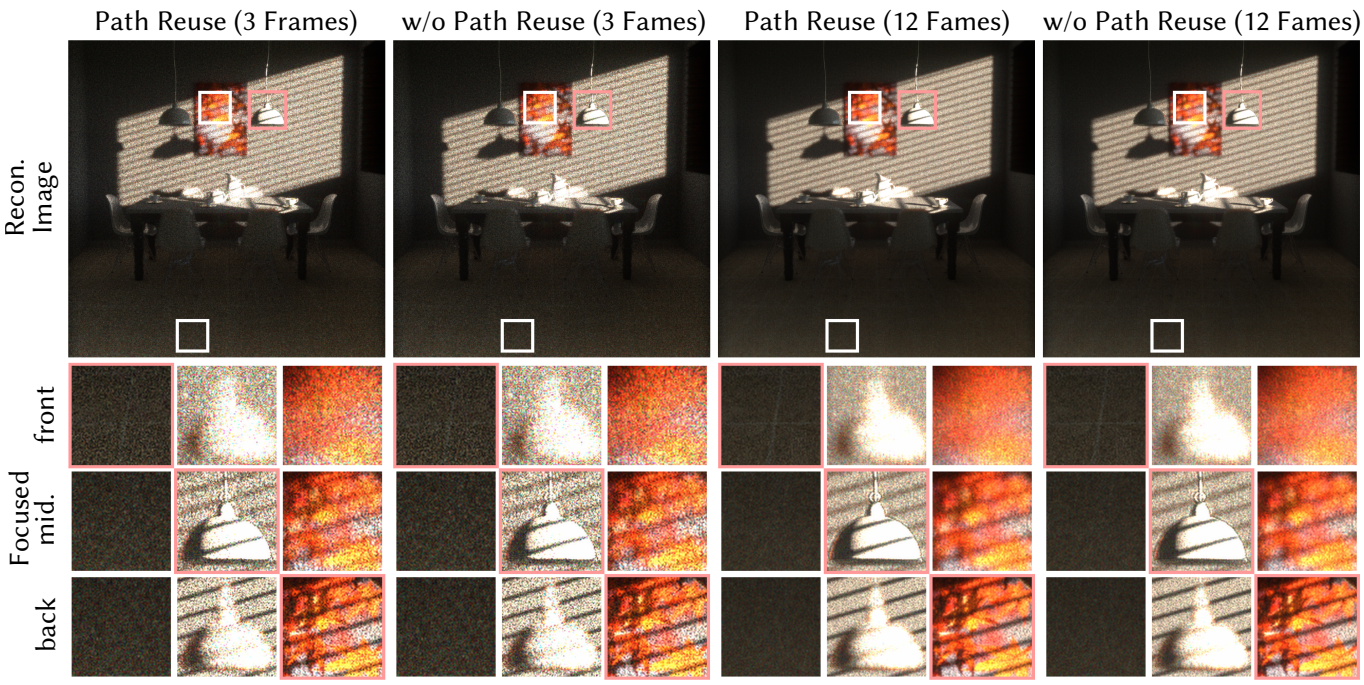


Fig. S1. **Reconstruction results with and without Path Reuse** under 3 and 12 frames. For each set, we present close-up regions with the viewpoint focused at front, middle, and back. The red box indicates in-focus region, while the white box indicates out-of-focus region. No noticeable differences can be observed between the two sets, validating that path reuse does not introduce noticeable correlation across frames nor reduce path diversity.

way, low-roughness surfaces can still preserve certain chance of coherent propagation and thus retain part of their imaging capability, while rougher surfaces are increasingly dominated by incoherent transport. The other is to delay the incoherent radiance tracing and continue coherent wave tracing through one or two additional non-δ interactions, so that a higher-order approximation can be achieved for low-roughness materials.

S1.3.2 Distances of recording plane. A second source of bias comes from the use of a single wave recording plane (WRP). The color drift observed for objects very close to the WRP (e.g. the chair back in the classroom scene) is a consequence of this recording model and its associated sampling tradeoff. The wave facet representation relies on a sufficient propagation distance to encode view dependency on the recording plane. If the WRP is placed too close to the scene, the available spatial resolution becomes insufficient to faithfully capture the local angular variation of near objects, leading to color drift. If it is placed too far, however, the wave-propagation sampling requirement increases substantially and the computation becomes much more expensive to converge. The current system therefore adopts a single WRP at a compromise distance. Extending the framework to multiple WRPs would be a principled way to better capture both near and far view-dependent structure while maintaining reasonable convergence.

S1.4 Details of Mesh-Baking Process

Monte Carlo ray tracing often exhibits significant stochastic noise at low sample counts, particularly on Lambertian (diffuse) surfaces, which demand a high number of samples per pixel to obtain a converged, noise-free estimate. Such high sampling density imposes a prohibitive computational overhead for real-time applications. Given that our experimental environments in this work are static, we employ a lightmap baking [Mitchell et al. 2006] strategy to pre-compute the diffuse illumination and store it in textures.

Mathematically, for a perfectly diffuse surface at point p, the outgoing radiance L_o in direction ω_o is defined by the hemispherical integral of the incoming radiance L_i weighted by the bidirectional reflectance distribution function (BRDF), expressed as:

$$L_o(p, \omega_o) = \int_\Omega f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) \cos\theta_i d\omega_i. \tag{S5}$$

For a Lambertian surface, the BRDF is a constant $f_r = \frac{\rho}{\pi}$, where ρ represents the surface albedo. As the outgoing radiance is uniform across all directions, the above equation can be simplified as:

$$L_o(p) = \frac{\rho(p)}{\pi} \int_\Omega L_i(p, \omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i. \tag{S6}$$

Thus, we pre-compute the exitant radiance (or irradiance) for all diffuse surfaces and store these values in 2K resolution lightmap textures via UV unwrapping. During online rendering, the diffuse lighting at a hit point is then efficiently retrieved via a simple texture lookup.

To balance computational performance and visual fidelity, we adapt our lookup strategy based on scene complexity, as follows:

- **Simple Scenes:** The baked value is queried directly at the first diffuse intersection point.

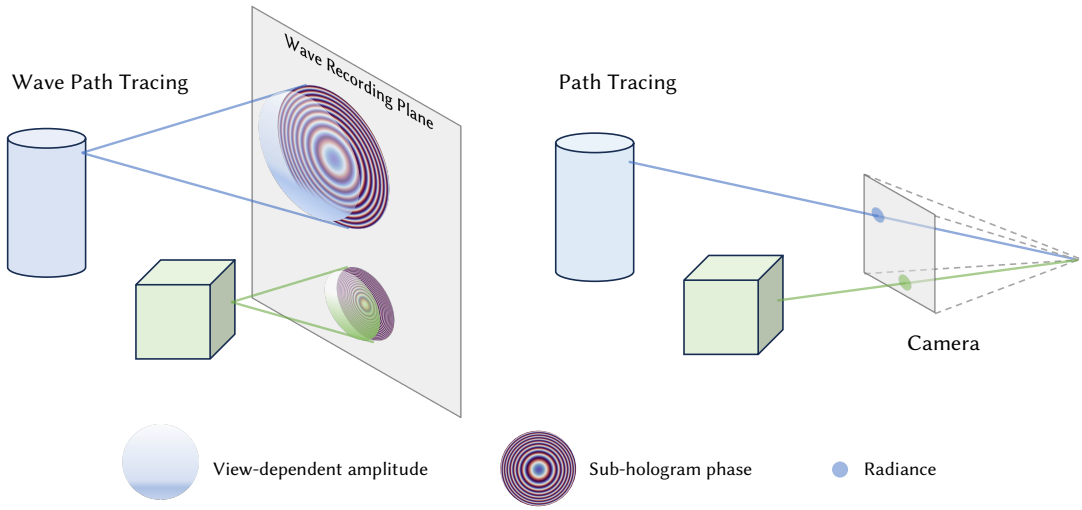


Fig. S2. **Per-sample Contributions in HoloPathTracer.** Left: in wave path tracing, each scene point contributes through a cone of rays to the wave recording plane, producing a complex-valued sub-hologram with view-dependent amplitude envelope. Right: in standard path tracing, each scene point contributes only real-valued radiance along a single camera ray. This complex-valued recording enables HoloPathTracer to reconstruct DoF images focused at arbitrary depths.

- **Complex Scenes:** Due to the finite resolution of the lightmaps, direct lookups at the first diffuse point can introduce noticeable artifacts near UV island boundaries or geometric seams. To mitigate this, we employ the baked textures at the second diffuse bounce. In this setting, the lightmap still effectively acts as a smoothed approximation of the indirect illumination, substantially reducing variance while maintaining high performance.

S1.5 Path Reuse for Time-Multiplexed Hologram Generation

Path Reuse is the key to enabling efficient generation of multiple time-multiplexed holograms with different random phase distributions under a path tracing framework. This does not introduce noticeable correlation as the paths and terms $e^{jd_{OPL}/\lambda}$ are in low frequency/variance compared with random phases, so such variance will not significantly affect the correlation between different frames. Nor does it reduce path diversity across frames, since path sampling only determines ray–object interactions and encodes the 3D spatial-angular relationships used for wavefield accumulation with physically correct visual cues. Consequently, different frames do not require different sampled paths.

To further clarify this point, we compare the reconstruction results using the same paths and using different paths for different frames in Figure S1.

S1.6 Visualization of Per-sample Contributions in HoloPathTracer

In the proposed HoloPathTracer, each ray sample contributes a complex value to the wave recording plane, and each wave facet can contribute through a cone of rays reaching that plane. Rays from nearby scene points collectively form a complex wavefront with view-dependent amplitude and a sub-hologram phase pattern, as shown in Figure S2. In contrast, standard path tracing accumulates only real-valued radiance, where each scene point contributes only along a single ray reaching the camera.

Consequently, standard path tracing records only an all-in-focus image or a DoF image focused at a fixed depth, whereas HoloPathTracer uses a complex-valued film to record interference patterns, enabling DoF reconstruction at arbitrary depths.

S2 Technical Overview of Baseline Methods

In the main manuscript, we compare our method with alternative CGH frameworks including the RGBD, focal stack, light field, mesh-based, and Gaussian splatting-based methods. Below, we provide further details regarding the implementation of these baseline methods.

RGBD. This is implemented based on the "*Neural 3D Holography*" [Choi et al. 2021]. Specifically, the input scene is partitioned into **4** distinct depth layers, with corresponding masks generated for each. For hologram encoding, a phase-only hologram is propagated to these 4 depths, masked by their corresponding masks, and evaluated with the masked target RGB image to compute loss. The optimization is then performed using an SGD-based optimization algorithm.

Focal Stack. This is implemented based on the "*Time-Multiplexed Neural Holography*" [Choi et al. 2022]. Similar to the RGBD counterpart, a phase-only hologram is propagated to **4** depth layers and optimized against the target image using a SGD-based optimization algorithm. However, in this case, the target image used for loss computation is pre-rendered with by a camera with finite aperture, yielding artificial defocus blur consistent with the depth-of-field formula.

Light Field. This is implemented based on the "*Pupil-Aware Holography*" [Chakravarthula et al. 2022]. A phase-only hologram is first

Fig. S3. **Illustration of 3D scenes** used in our implementation.

propagated using ASM to different depth planes and filtered in the Fourier plane corresponding to multiple viewpoints. The resulting viewpoints' complex wavefronts are then evaluated with the focal stack of the corresponding view to compute the loss. In our implementation, we use a **6×6** grid of viewpoints for the light field. Each viewpoint is associated with a focal stack comprising **4** depth layers. During the SGD-based optimization, a focal stack from a randomly selected viewpoint is set as the target at each iteration to save memory.

Mesh-based. This is implemented based on *the work by Matsushima* et al. [2003], who propose the spectrum remapping to achieve tilted ASM propagation for polygon (mesh)-based CGH. In our implementation, each polygonal facet of the mesh is propagated via the tilted ASM to the recording plane, yielding a complex wave field, which can be further encoded into a phase-only hologram using complex amplitude-supervised SGD (see Sec. S3.2).

Gaussian Splatting. This is implemented based on available open-source code of "*Gaussian Wave Splatting Holography*" [Choi et al. 2025]. Specifically, we propagate the GS representation of a 3D scene to the recording plane to obtain the complex wave field, which can be further encoded into a phase-only hologram using complex amplitude-supervised SGD (see Sec. S3.2).

S3 Hologram Synthesis

S3.1 Time-multiplexing Optimization

To obtain holograms with different random phase distributions for time-multiplexed holography, FFT-based propagation methods—including RGB-D, focal stack, light field, and mesh-based approaches—require reassigning random phases, which leads to a nearly linear growth in computational overhead. "Gaussian Wave Splatting Holography [Choi et al. 2025]" does not support assigning random phases to the wave field. Our method enables flexible assignment of random phases to the generated holograms with negligible additional computation time (refer also back to Fig. 6 in the main manuscript).

S3.2 Hologram Encoding Elaboration

We build a holographic display prototype for validation, which employs a phase-only SLM to load the hologram. Consequently, we use the complex amplitude-supervised, SGD-based optimization to encode the complex hologram into a phase-only pattern for display. To account for phase shifting invariance and to normalize the field energy, we further apply a complex-valued scale factor s to match the reconstruction with the target. This s can be computed as:

$$\begin{aligned} \mathrm{Re}(s) &= \frac{\sum(\mathrm{Re}\{u_\mathrm{r}\}\cdot\mathrm{Re}\{u_\mathrm{t}\}+\mathrm{Im}\{u_\mathrm{r}\}\cdot\mathrm{Im}\{u_\mathrm{t}\})}{\sum(\mathrm{Re}\{u_\mathrm{r}\}^2+\mathrm{Im}\{u_\mathrm{r}\}^2)}, \\ \mathrm{Im}(s) &= \frac{\sum(-\mathrm{Im}\{u_\mathrm{r}\}\cdot\mathrm{Re}\{u_\mathrm{t}\}+\mathrm{Re}\{u_\mathrm{r}\}\cdot\mathrm{Im}\{u_\mathrm{t}\})}{\sum(\mathrm{Re}\{u_\mathrm{r}\}^2+\mathrm{Im}\{u_\mathrm{r}\}^2)}, \end{aligned} \tag{S7}$$

where u_r and u_t represent the reconstructed wave field and the target wave field, respectively. Subsequently, the optimization is guided by the loss evaluated between $s \cdot u_r$ and u_t.

Notably, as this encoding optimization requires supervision only on the complex wave field of one single depth plane, it is much faster than conventional multi-depth- or multi-view-supervised encoding methods. The computational overhead of supervising multiple depths increases almost linearly with the number of depth planes. For example, supervising 4 depth planes with a vanilla SGD algorithm (e.g., for RGBD/focal stack) requires approximately 4× the runtime compared to our complex amplitude-supervised SGD variant. We show a comparison of the reconstruction results between complex hologram and phase-only hologram in Fig. S6. And quantitative assessment in Table S2. The phase-only hologram shows some contrast loss due to its reduced modulation degrees of freedom, but it still preserves the core visual cues and depth-dependent effects.

S4 Illustration of Representative 3D Scenes for Our Rendering

We visualize the 3D scenes [Bitterli 2016] used for our wave field rendering in Fig. S3.

S5 Bench-top Holographic Display and Additional Results

S5.1 Experimental Setup and Specifications

Our experimental setup is illustrated in Fig. S4, with specifications of the employed devices and parts listed in Table S1.

S5.2 Calibration Details

Before using the SLM for holographic display, it is essential to characterize its phase modulation response as a function of the displayed gray value, i.e., the look-up table (LUT). Although the LUT is typically provided by the manufacturer, it can vary with environmental conditions and over time. In this work, we recalibrate the LUT using self-referenced interferometry [Zhao et al. 2018]. We present the calibration patterns and corresponding captured results of the reconstruction in Fig. S5.

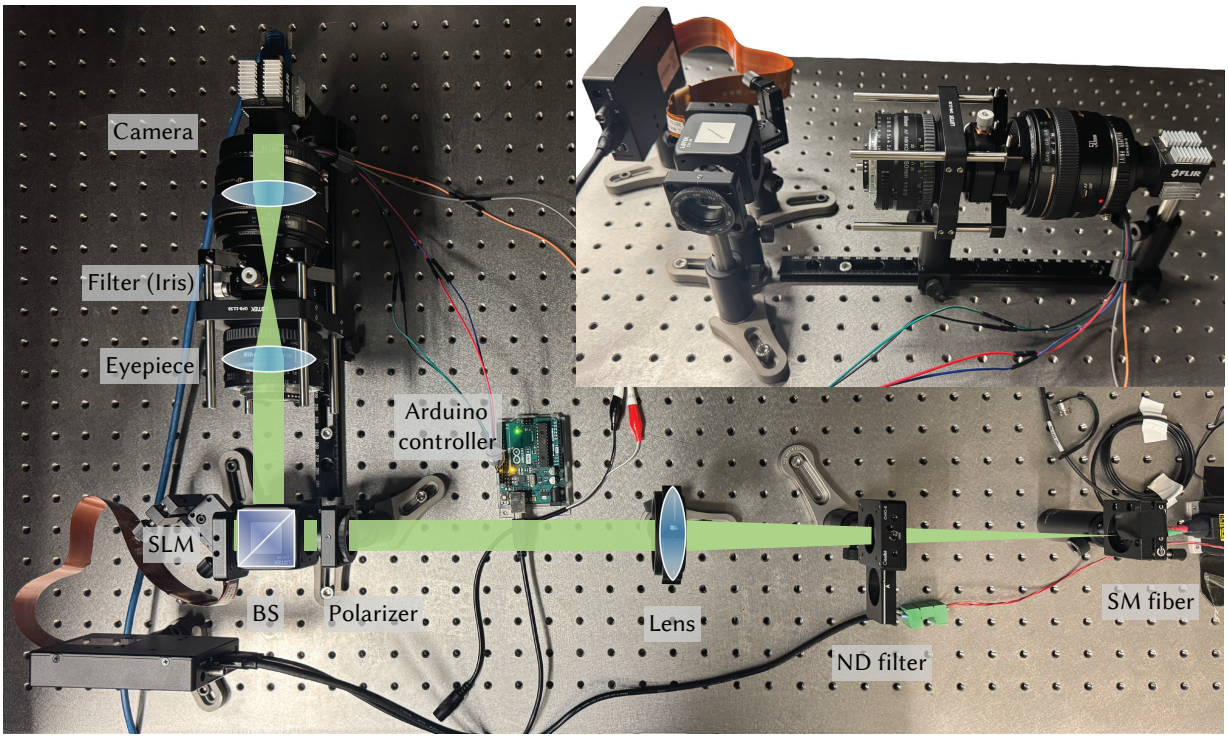


Fig. S4. **Photograph of our holographic display prototype**. The light emitted from the single-mode fiber-coupled RGB laser source is first collimated by a lens and then directed onto the SLM through a polarizer and a beam splitter (BS). The modulated light from the SLM is reflected by the beam splitter and relayed to the sensor through an eyepiece and a camera lens. Notably, the polarizer ensures the polarization direction of the incident light on the SLM matches the orientation of the liquid crystal molecules for optimal phase modulation. A frequency filter is placed between the eyepiece and the camera lens to block stray light and unwanted diffraction orders. We use an Arduino micro-controller to tune the focus distance of the camera lens. The top-right inset shows a close-up view of the light engine and capture module.

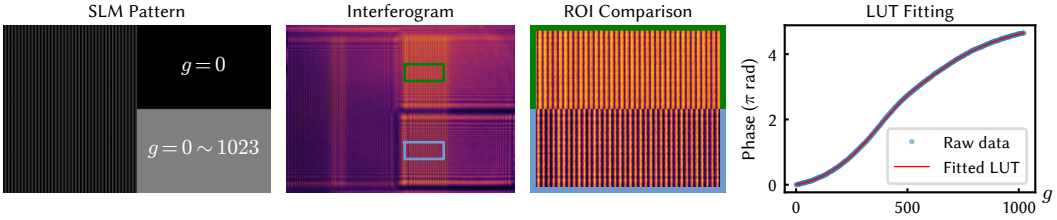


Fig. S5. **Illustration of SLM calibration**. The left-most column illustrates the calibration pattern displayed on the SLM. The center two columns denote the captured interferometric fringes and the region of interest (ROI) in reference and measurement areas, respectively. The right-most column illustrates the extracted phase values and the corresponding fitted LUT curve. Note, we illustrate calibration results for red channel only for brevity; green and blue channels are calibrated in a similar manner.

Table S1. **Specifications of the devices and parts** used to build our holographic display prototype.

Parts	Models & Parameters
SLM	HDSLM80R Plus (UPOLabs, 1,920 × 1,200, 10 bit)
RGB Laser	Fisbar ReadyBeam (640 nm, 516 nm, 455 nm)
Sensor	Grasshopper 3 (FLIR, 4,096 × 3,000, pixel pitch 3.45 µm)
Collimating Lens	f = 300 mm (Achromatic doublet)
Eyepiece lens	f = 50 mm (Nikon SLR lens)
Camera lens	f = 50 mm (Canon SLR lens)
Beam Splitter	400–700 nm, Non-polarized, 50:50 (R:T)

Specifically, we split the SLM plane into three regions, where a half plane displays a binary grating pattern, and the other two quarters display uniform gray levels. One of the two quarters serves as the reference region with a fixed zero gray level, while the other quarter sequentially displays gray levels from 0 to $2^{10} - 1$ (10-bit SLM) to cover the full modulation range. The binary grating pattern deflects half area of the incident beam, which interferes with the sub-beams reflected from the two uniform regions, forming two sets of interferometric fringes on the observation plane. During calibration, we capture 256 interferograms corresponding to 256

gray levels displayed on the measurement region. For LUT calibration, we are interested in extracting the phase difference between the reference and measurement regions, which is encoded in the relative position shift of the interferometric fringes. To this end, we first select two ROIs for the reference and measurement areas, respectively. Then, we extract the phase difference between the two fringe patterns using Fourier transform. Finally, we fit the extracted phase differences using bicubic spline interpolation to reduce the noise and obtain the LUT curve. With the LUT curve, we can query the required gray level by interpolation to achieve a target phase modulation.

S5.3 Additional Results

We present additional simulation results in Fig. S7, as well as quantitative assessment in PSNR (dB) of all rendering results (5 scenes) in Table S2.

Specifically, to faithfully evaluate the 3D visual realism (including both focal stack and light field aspects) of holographic reconstructions, the reported PSNR is computed as the *average* of two components:

1) the PSNR of the center-view focal-stack (5 focused depths);

2) the average of PSNRs of all 7×7 views' focal-stacks (5 focused depths).

S6 Supplementary Video

See also the supplementary video `HoloPathTracer.mp4` for elaborated analysis and results.

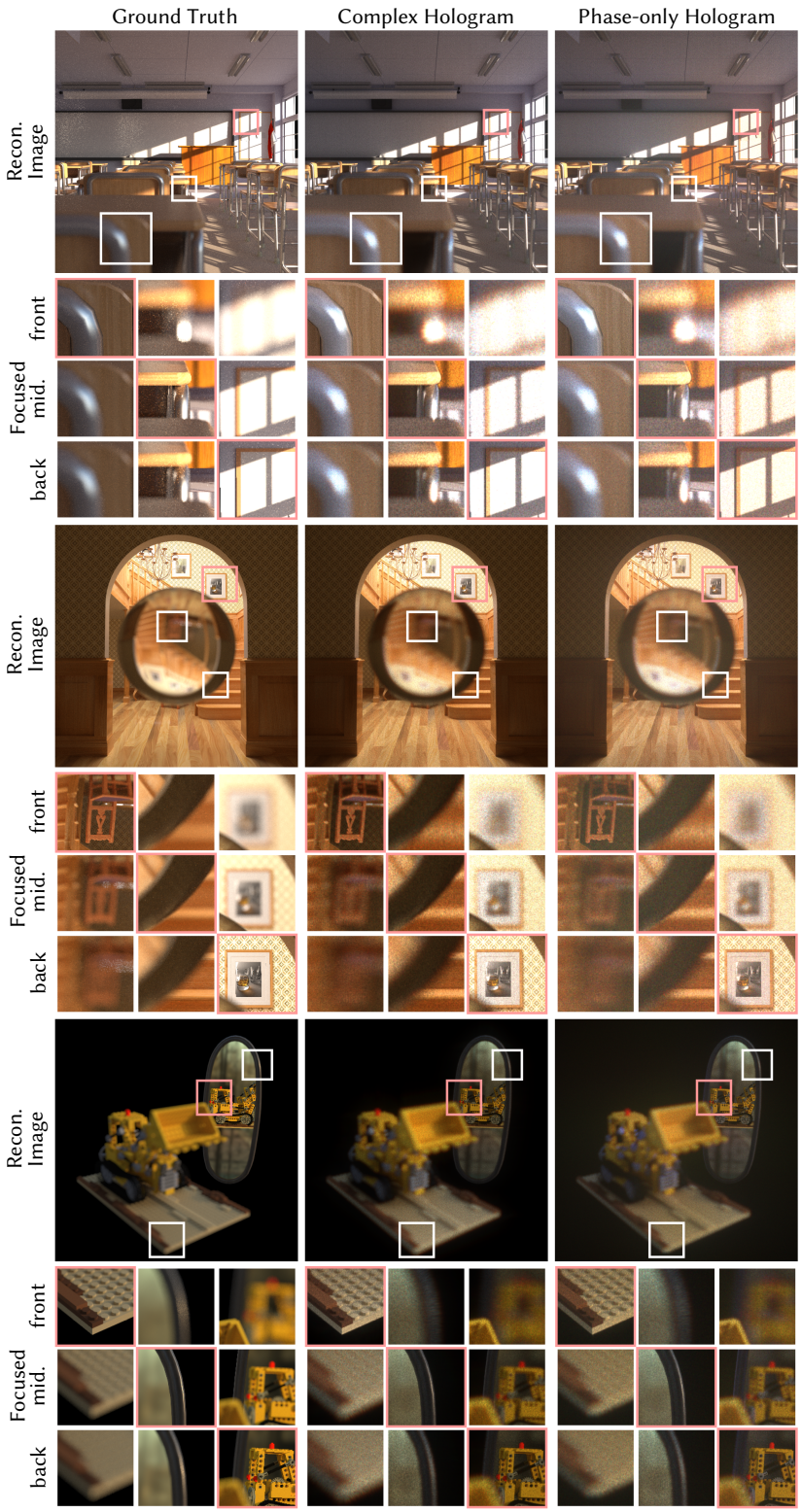


Fig. S6. **Reconstruction results comparison between complex hologram and phase-only hologram**. For each set, we present close-up regions with the viewpoint focused at front, middle, and back. Red box indicates in-focus region, while white box indicates out-of-focus region. Phase-only holograms show some contrast loss due to their reduced modulation degrees of freedom.

Table S2. **Quantitative assessment of rendering results**. We report (average) PSNR (dB) ↑ metrics for multi-view image stacks rendered from holographic reconstructions across various CGH frameworks. Notably, GWS cannot support wave field rendering for highly complex scenes; and PSNR values for Mesh results fall far below the threshold for meaningful quality. For our methods, both metrics from complex holograms and phase-only holograms are reported.

Scene	Ours (full) [phase-only]	Ours (fast) [phase-only]	RGBD	FS	LF	GWS	Mesh
Bathroom	20.13 [18.44]	13.16 [12.39]	9.21	12.38	12.72	—	—
Teapot	14.40 [14.26]	14.38 [14.03]	7.69	7.05	7.96	11.57	—
Lego	29.03 [28.17]	28.75 [27.94]	21.09	23.47	21.90	22.99	—
Staircase	23.69 [22.03]	22.67 [20.75]	13.14	13.84	16.37	—	—
Classroom	18.12 [16.78]	19.50 [18.49]	11.29	12.97	14.83	—	—

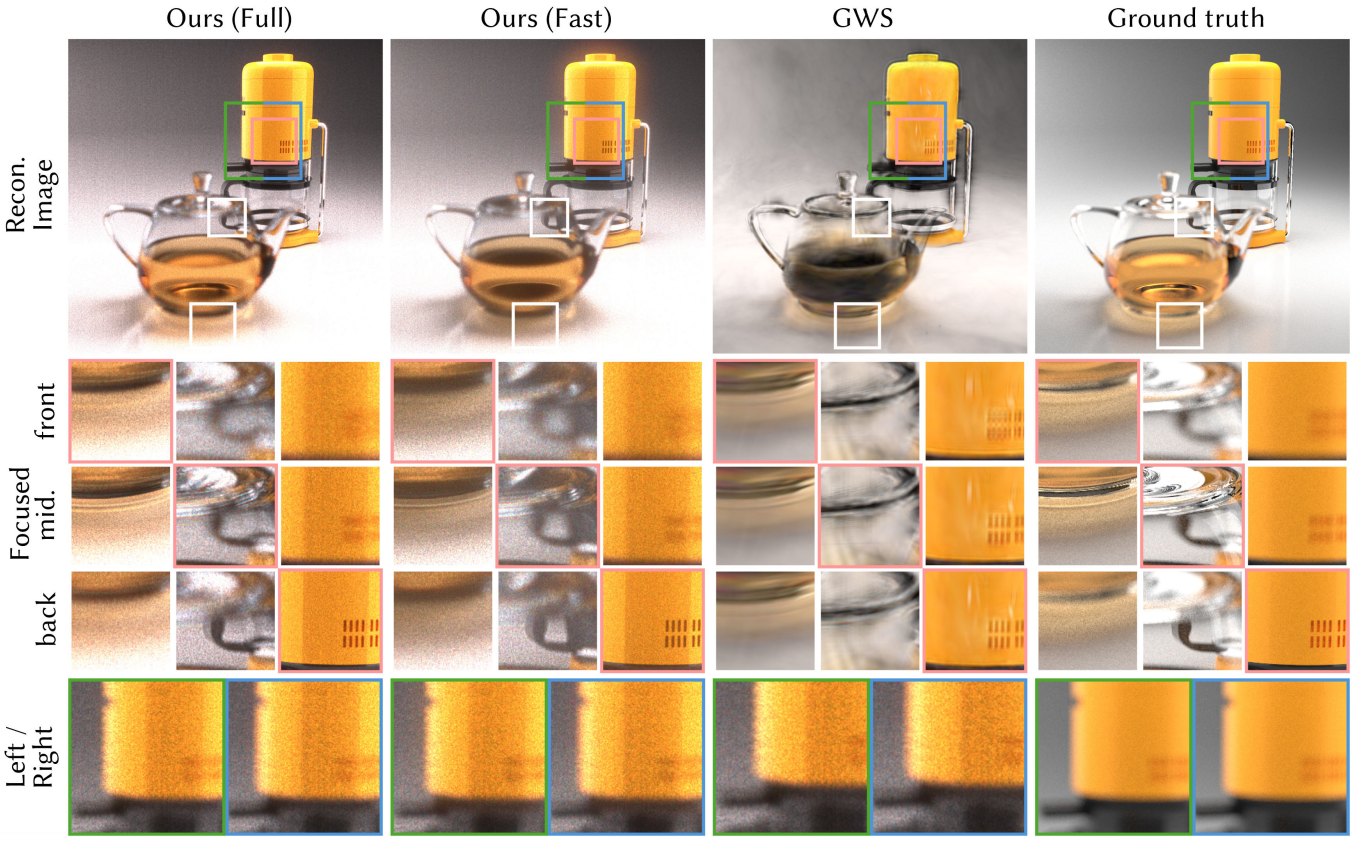


Fig. S7. **Additional simulation results** of our two variants, GWS, and GT. For each set, we present close-up regions with the viewpoint focused at front, middle, and back, as well as the left/right view-dependent effects.

References

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Praneeth Chakravarthula, Seung-Hwan Baek, Florian Schiffers, Ethan Tseng, Grace Kuo, Andrew Maimone, Nathan Matsuda, Oliver Cossairt, Douglas Lanman, and Felix Heide. 2022. Pupil-Aware Holography. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–15. doi:10.1145/3550454.3555508

Suyeon Choi, Brian Chao, Jacqueline Yang, Manu Gopakumar, and Gordon Wetzstein. 2025. Gaussian wave splatting for computer-generated holography. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–13. doi:10.1145/3731163

Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, Matthew O'Toole, and Gordon Wetzstein. 2022. Time-multiplexed neural holography: a flexible framework for holographic near-eye displays with fast heavily-quantized spatial light modulators. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9. doi:10.1145/3528233.3530734

Suyeon Choi, Manu Gopakumar, Yifan Peng, Jonghyun Kim, and Gordon Wetzstein. 2021. Neural 3D Holography: Learning Accurate Wave Propagation Models for 3D Holographic Virtual and Augmented Reality Displays. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–12. doi:10.1145/3478513.3480542

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

Kyoji Matsushima, Hagen Schimmel, and Frank Wyrowski. 2003. Fast calculation method for optical diffraction on tilted planes by use of the angular spectrum of plane waves. *Journal of the Optical Society of America A* 20, 9 (2003), 1755–1762. doi:10.1364/JOSAA.20.001755

Kyoji Matsushima and Tomoyoshi Shimobaba. 2009. Band-Limited Angular Spectrum Method for Numerical Simulation of Free-Space Propagation in Far and Near Fields. *Opt. Express* 17, 22 (2009), 19662–19673. doi:10.1364/OE.17.019662

Jason Mitchell, Gary McTaggart, and Chris Green. 2006. Shading in valve's source engine. In *ACM SIGGRAPH 2006 Courses* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 129–142. doi:10.1145/1185657.1185832

Zixin Zhao, Zhaoxian Xiao, Yiying Zhuang, Hangying Zhang, and Hong Zhao. 2018. An interferometric method for local phase modulation calibration of LC-SLM using self-generated phase grating. *Review of scientific instruments* 89, 8 (2018), 083116. doi:10.1063/1.5031938