

# Complex-Valued Holographic Radiance Fields

YICHENG ZHAN, University College London, United Kingdom

DONG-HA SHIN, Pohang University of Science and Technology (POSTECH), South Korea

SEUNG-HWAN BAEK, Pohang University of Science and Technology (POSTECH), South Korea

KAAN AKŞIT, University College London, United Kingdom

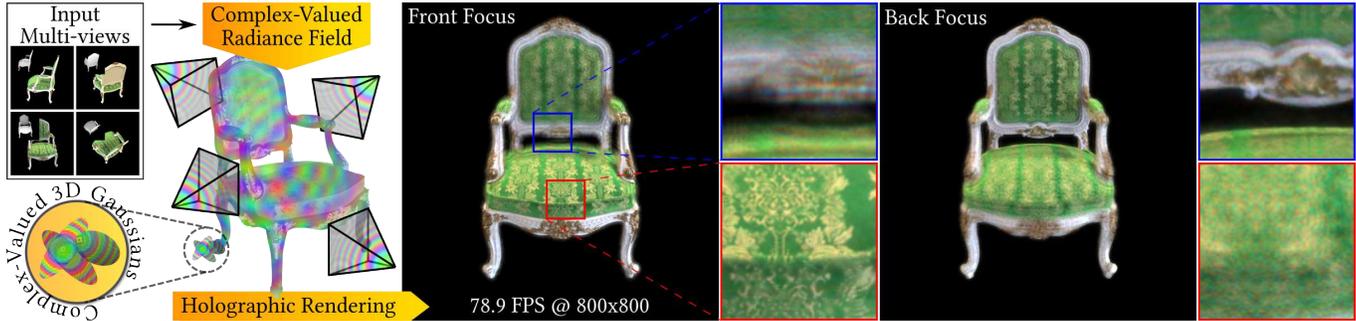


Fig. 1. Complex-Valued Holographic Radiance Fields. Our method explicitly treats the complex-valued holographic radiance fields as the primary optimization target, jointly optimizing the 3D scene across multiple viewpoints to model intensity, interference, and diffraction of light in scene representations. The complex-valued 3D Gaussians will be projected to multiple depth planes, resulting in a complex 3D hologram via holographic rendering. This enables holographic reconstruction with depth-of-field effects: blue boxes highlight the back focus region, and red boxes show the front focus region.

Modeling wave properties of light is an important milestone for advancing physically-based rendering. In this paper, we propose complex-valued holographic radiance fields, a method that optimizes scenes without relying on intensity-based intermediaries. By leveraging multi-view images, our method directly optimizes a scene representation using complex-valued Gaussian primitives representing amplitude and phase values aligned with the scene geometry. Our approach eliminates the need for computationally expensive holographic rendering that typically utilizes a single view of a given scene. This accelerates holographic rendering speed by 30x-10,000x while achieving on-par image quality with state-of-the-art holography methods, representing a promising step towards bridging the representation gap between modeling wave properties of light and 3D geometry of scenes.

CCS Concepts: • **Computing methodologies** → **Mixed / augmented reality; 3D imaging; Computer graphics; Rendering.**

Additional Key Words and Phrases: Novel View Synthesis, Radiance Fields, 3D Gaussians, Computer-Generated Holography

Authors' addresses: Yicheng Zhan, UCABY83@ucl.ac.uk, University College London, Gower Street, London, United Kingdom, WC1E 6BT; Dong-Ha Shin, 0218sdh@gmail.com, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro, Nam-gu, Pohang-si, South Korea, 37673; Seung-Hwan Baek, shwbaek@postech.ac.kr, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro, Nam-gu, Pohang-si, South Korea, 37673; Kaan Akşit, University College London, Gower Street, London, United Kingdom, k.aksit@ucl.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Association for Computing Machinery.

XXXX-XXXX/2024/3-ART \$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

## ACM Reference Format:

Yicheng Zhan, Dong-Ha Shin, Seung-Hwan Baek, and Kaan Akşit. 2024. Complex-Valued Holographic Radiance Fields. 1, 1 (March 2024), 17 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 Introduction

Light exhibits numerous physical phenomena beyond mere intensity, including wave-optics properties such as spectral variations [Kim et al. 2023], polarization [Zhao et al. 2022], diffraction, and interference [Goodman 2005]. Accounting for wave-optics properties of light more accurately could help increase the visual fidelity in 3D scene representations [Wen et al. 2025]. Most recently, 3D Gaussian splatting (3DGS) [Kerbl et al. 2023] has emerged as an effective 3D scene representation to synthesize novel-view intensity images. However, 3DGS neglects wave-optics properties of light. In addition, the applications of physically-based rendering and emerging holographic 3D displays [Kim et al. 2024] in computer graphics demand modeling these wave-optics properties to be accounted for.

computer-generated holography (CGH) synthesizes holograms that faithfully reconstruct wavefronts emitted from 3D scenes. While CGH has been predominantly utilized for holographic display applications, it has recently garnered increasing interest from the rendering community for generating physically-plausible scenes in computer graphics [Steinberg et al. 2024, 2022]. Existing CGH methods treat hologram synthesis as a post-processing task applied after the conventional rendering pipeline. These methods typically estimate holograms that reproduce the desired reconstructed intensity image for a fixed viewpoint [Shi et al. 2021] or a limited viewing angle [Chakravarthula et al. 2022; Chen et al. 2025]. When such methods are queried with a new viewpoint, they have to be recalculated for a new hologram using only the provided viewpoint

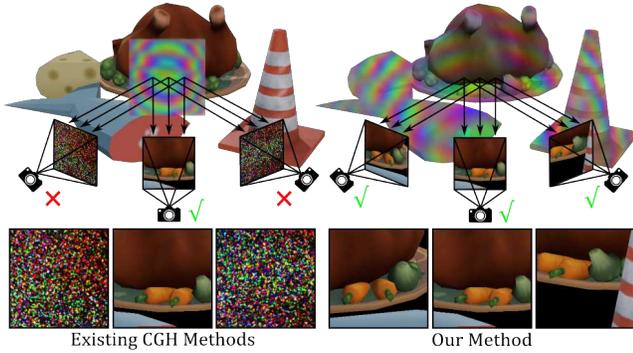


Fig. 2. Left: Existing CGH methods do not preserve geometry of scenes under viewpoint changes, necessitating transformation or recalculation per-view. Right: Our complex-valued holographic radiance field offers a 3D consistent representation in contrast.

information. Most recently, the work by Choi et al. [2025] generates holograms of 3D scenes represented with real-valued, intensity-based 2D Gaussian primitives. Their method introduces multi-view based approaches tailored for CGH for the first time, requiring an additional step of transformation for each queried viewpoint, greatly increasing computational complexity. These methods adopt an *Eulerian* paradigm, in which the hologram plane remains fixed while the complex field is recomputed at each spatial coordinate. In this context, **previous CGH methods lack persistent geometric representations, computing holograms at the camera plane rather than coupling them to 3D scene geometry**. Thus, such holographic scene representations without any transformation or recalculation would lead to noise-like visuals rather than perspective images when the viewpoint changes, as demonstrated in Fig. 2.

Our work aims to merge 3DGS radiance fields and holographic rendering under one umbrella using complex numbers, which are widely adopted in holography for representing both amplitude and phase information from 3D scenes [Goodman 2005]. In this paper, we first introduce complex-valued Gaussian primitives. Our scene representation framework utilizes these complex-valued Gaussian primitives to model both amplitude and phase emitted from 3D scenes. Unlike previous intensity-based modeling methods [Choi et al. 2025], our method explicitly treats complex-valued holographic radiance fields as the primary optimization target, without the need for per-view transformation [Choi et al. 2025] or fixed viewpoint hologram recalculation [Chen et al. 2025]. Secondly, in our framework the simulation of diffraction and interference phenomena is formulated as a differentiable rasterizer, rendering complex-valued 3D Gaussian primitives from novel views. In contrast to the existing CGH methods [Choi et al. 2025] that scale quadratically with the number of primitives, our approach achieves a linear relationship between the primitives and complexity, achieving 30x-10,000x speed improvements. By optimizing directly in the complex-valued domain across multiple viewpoints, our method achieves view-dependent diffraction and interference effects without relying on intensity-based intermediaries. Specifically, our contributions include:

- Complex-valued holographic radiance fields. We present a new variant of 3DGS, utilizing complex-valued Gaussian primitives. Uniquely, our method is transformation-free and follows the geometry of 3D scenes in terms of the phase and amplitude, regardless of the viewpoints.
- Complex-valued differentiable wave-optics rasterizer. We enabled efficient rendering and optimization of complex-valued holographic radiance fields by rasterizing optical waves emitted from a 3D scene.

We evaluate the image quality and speed of our method by comparing against existing Gaussian-based CGH approaches. Our codebase is publicly available at our project page <sup>1</sup>.

## 2 Related Work

Our approach bridges recent advances in neural rendering with holography, enabling the efficient generation of holograms from novel views. We will first review novel-view synthesis methods and then examine existing wave optics rendering techniques that also attempt to simulate light phenomena beyond intensity. Finally, we discuss the traditional CGH methods in general, including point-based, polygon-based, and layer-based methods.

### 2.1 Novel-View Synthesis

Traditional novel-view synthesis methods primarily model intensity and appearance, limiting their utility for emerging holographic display technologies that require phase-aware representations. Early techniques such as image-based rendering [Chaurasia et al. 2013] and volumetric scene representations [Hedman et al. 2017; Riegler and Koltun 2020] laid the groundwork for view-dependent appearance modeling. Building on these foundations, neural radiance fields (NeRF) [Mildenhall et al. 2021] advanced the field by learning continuous radiance fields that capture view-dependent color and opacity. This representation has since been extended to improve visual quality and computational efficiency [Barron et al. 2021, 2022, 2023; Chen et al. 2022; Duckworth et al. 2024; Niemeyer et al. 2024; Reibain et al. 2021].

Point-based representations have recently advanced with 3DGS [Kerbl et al. 2023], emerging as a promising alternative to NeRF. 3DGS offers a promising foundation for our work, as it encodes both geometry and appearance using oriented Gaussian primitives, but it is still fundamentally limited to intensity-based representations. Our work extends 3DGS to incorporate intrinsic amplitude and phase information, enabling it to model complex-valued holographic radiance fields that capture not only intensity but also interference and diffraction phenomena.

*Rendering Beyond Intensity.* The work by Kim et al. [2023] extends NeRF beyond intensity-based representations to capture spectral and polarimetric properties of light, revealing richer material and structural characteristics of 3D scenes. Notably, spectral and polarization data can be supervised using ground-truth measurements obtained from specialized imaging devices. On the other hand, interference and diffraction phenomena arise from the superposition of phase and amplitude values. Interference and diffraction

<sup>1</sup>[https://github.com/complight/Complex\\_Valued\\_Holographic\\_Radiance\\_Fields](https://github.com/complight/Complex_Valued_Holographic_Radiance_Fields)

effects lack direct measurement data and must be inferred through computational means, as holographic cameras capable of capturing both amplitude and phase information are not yet commodity equipment [An et al. 2020]. Moreover, supporting simulations of holography in rendering pipelines is a common theme in research. For instance, Steinberg et al. [2024; 2022] introduced wave optics formulations for path tracing, enabling partial coherence and diffraction effects. Choi et al. [2025] proposed Gaussian Wave Splatting (GWS) that transforms 2D Gaussian Splatting [Huang et al. 2024] scenes into holograms by deriving closed-form solutions for angular spectrum. Chen et al. [2025] proposed a two-stage method that combines pretrained intensity-based 3DGS with U-Net [Ronneberger et al. 2015], enabling efficient hologram synthesis for novel views. In this paper, we propose to enable recalculation-free novel-view hologram synthesis, whereas Choi et al. requires per-view wave re-computation and Chen et al. requires per-view U-Net inference.

## 2.2 Eulerian-based Computer-Generated Holography

Holographic displays rely on CGH, which algorithmically synthesizes diffraction and interference patterns to reconstruct 3D scenes with near-accurate depth cues. *Conceptually, traditional CGH methods follow a per-view Eulerian-based approach to wave propagation*, where the coordinates on the hologram plane remain fixed, while the complex field varies when the relationship between the scene geometry and the viewpoint changes, requiring expensive recalculation. The computed holographic patterns are implemented using spatial light modulator (SLM), which modulates the phase of incident coherent beams to generate designed image patterns. Through numerical wave propagation simulations [Kavaklı et al. 2022; Peng et al. 2020], CGH generates holograms leading to high spatio-angular resolution, positioning it as a promising emerging screen technology. The accuracy and computational complexity of CGH strongly depend on primitive representations, which are generally categorized into point-based, polygon-based, and layer-based methods.

Point-based CGH represents scenes using discrete points emitting spherical waves. The wave field at the hologram,  $P$ , is calculated by summing waves from  $N$  points

$$P = \sum_{n=1}^N A_n \frac{e^{jkr_n}}{r_n}, \quad (1)$$

where  $A_n$  denotes amplitude,  $k$  is the wave number, and  $r_n$  is the propagation distance from point  $n$  to  $P(x, y)$ . Reflecting the Huygens-Fresnel principle, Eq. (1) is highly flexible and physically accurate. Maimone et al. [2017] presented point-source-based CGH computation for holographic near-eye displays for virtual and augmented reality. Shi et al. [2021] leveraged point clouds and triangle-ray intersection for occlusion handling in large-scale hologram datasets. However, the computational cost of point-based CGH increases significantly with point density, and occlusion handling introduces additional overhead.

Polygon-based CGH balances efficiency and accuracy by representing surfaces using triangular or polygonal facets emitting coherent wave fields. This method naturally integrates shading models and intrinsic occlusion handling through visibility algorithms (e.g. back-face culling and affine transformations) [Matsushima and

Nakahara 2009; Zhang et al. 2018b]. Mesh-based CGH includes topology information for precise occlusion handling [Ahrenberg et al. 2008; Yeom et al. 2022]. While achieving realistic shading by combining wave optics with rendering pipelines, these methods suffer from increased algorithmic complexity and struggle with finely-detailed or non-polygonal structures.

Layer-based CGH subdivides scenes into discrete depth planes, propagating complex fields from  $P$  independently to each plane using operators like the Angular Spectrum Method (ASM) [Matsushima and Shimobaba 2009; Zhang et al. 2020; Zheng et al. 2024]. The  $P(x, y)$  is optimized so the propagated results across all planes approximate the desired 3D scene. Layer-based CGH greatly accelerates computations through fast Fourier transform (FFT)-based propagation, making it suitable for faster rendering and is widely adopted in learned CGH methods [Choi et al. 2021; Liu et al. 2023; Peng et al. 2020; Shi et al. 2022; Zhan et al. 2024]. Building on this acceleration, learned complex-valued CGH methods [Zhang et al. 2025; Zhong et al. 2023] generate phase-only holograms from complex-valued fields in a single forward pass. These approaches share the same application as our method for SLM-based holographic displays. In contrast, our method embeds complex-valued primitives directly into the 3D scene representation, eliminating the need for re-inference per viewpoint.

While recent CGH advancements enable realistic defocus effects and continuous pupil accommodation [Kavaklı et al. 2023a; Kim et al. 2024], existing light field-based 3D and 4D CGH methods are typically optimized for limited viewing directions, with computed holograms losing the scene information under large viewpoint changes. We propose integrating strengths of both point-based and layer-based CGH methods, utilizing complex-valued 3D Gaussians as point-based primitives to encode scene geometry and complex-valued radiance. *Unlike existing Eulerian-based CGH approaches, our method follows a Lagrangian-based formulation*, treating amplitude and phase as intrinsic variables in scene representations. Compared with point clouds and polygon-meshes, Gaussians have learnable scale and rotation parameters, allowing them to capture anisotropic structures and fine-grained spatial variations effectively. This expressiveness enables both effective appearance modeling and geometrically faithful wave emitting distribution. Table 1 presents a comparison between our method and other Gaussian-based CGH approaches, demonstrating its ability to achieve fast inference, natural defocus blur, on-par image quality, and scene geometry-aware amplitude and phase representations across novel views.

## 3 Background and Problem Definition

Before explaining our approach, we first introduce the key concepts and terminology in this paper to define the research problem we are trying to solve. In this section, we explain several primary components to describe the research problem. These components are: 3D Gaussians for scene representations, hologram planes for wave recording, and camera parameters for view-dependent rendering.

*Representing 3D scenes with intensity-based Gaussians.* In line with the approach of [Kerbl et al. 2023], we learn geometry and radiance of scenes through the optimization of 3D Gaussian primitives. We consider a 3D region in Euclidean space,  $\Omega \subset \mathbb{R}^3$ . A scene inside

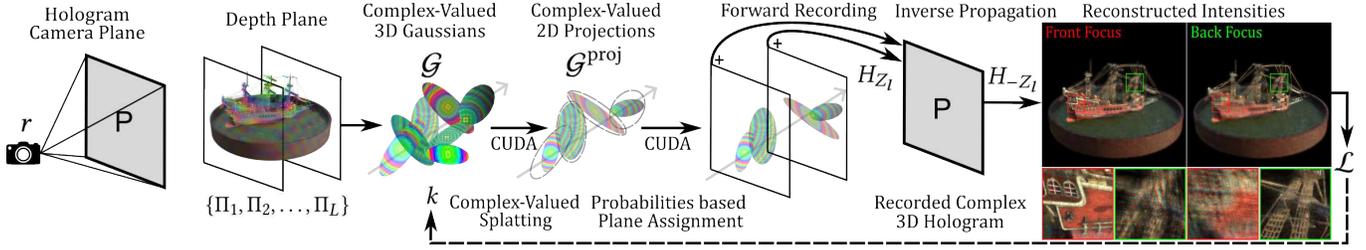


Fig. 3. Without the need to query an intensity-based radiance field, our approach models a complex-valued holographic radiance field using Gaussian primitives with intrinsic amplitude and phase properties, enabling scene geometry-aware amplitude and phase modeling across viewpoints and efficient rendering through a differentiable multi-layer propagation pipeline.

Table 1. Comparison of Gaussian primitives-based hologram synthesis method. *Inference* refers to inference speed. *Re* refers to whether the method requires recalculation when geometric relationship between scene and observer changes. *Type* refers to Gaussian representation type. *Quality* refers to the image quality of reconstructed images from the holograms. *Align* refers to scene geometry-aware amplitude and phase representations across novel views. *NDB* refers to natural defocus blur.

Method	Inference	Re	Type	Quality	Align	NDB
3DGS + U-Net [2025]	Fast	Yes	Intensity	Good	No	No
GWS [2025]	Moderate	Yes	Intensity	Good	No	No
Our Method	Fast	No	Complex	On-par	Yes	Yes

$\Omega$  is described using a set of  $N$  oriented 3D Gaussians, and each Gaussian is denoted by  $\mathcal{G}_n$ . Typically, each  $\mathcal{G}_n$  is parameterized by  $k = (\mathbf{c}_n, \mathbf{x}_n, \mathbf{R}_n, \mathbf{S}_n, \alpha_n)$ , where  $\mathbf{c}_n \in \mathbb{R}^3$  represents color,  $\mathbf{x}_n \in \mathbb{R}^3$  represents the center point of a Gaussian,  $\mathbf{R}_n \in \mathbb{R}^4$  describes the rotation of a Gaussian,  $\mathbf{S}_n$  represents scale along principal axes defined by  $\mathbf{R}_n$ , and  $\alpha_n$  represents the opacity. The *geometrical shape* of a 3D Gaussian is defined as

$$\mathcal{G}(\mathbf{x}, \mathbf{R}, \mathbf{S}) = \exp\left(-\frac{1}{2} \mathbf{x}^\top \Sigma^{-1} \mathbf{x}\right), \quad (2)$$

where the covariance,  $\Sigma$ , is decomposed into a rotation matrix  $\mathbf{R}$ , and scaling matrix  $\mathbf{S}$ , following

$$\Sigma = \mathbf{R} \mathbf{S} \mathbf{S}^\top \mathbf{R}^\top. \quad (3)$$

We project these 3D Gaussians onto 2D planes to render images at a camera plane, a process also known as the *differential splatting*. The process of differential splatting involves applying the transformation matrix,  $\mathbf{W}$ , and the Jacobian projection function,  $\mathbf{J}$ , mapping 3D points to 2D image plane. The covariance in camera-space,  $\Sigma'$ , is defined as

$$\Sigma' = \mathbf{J} \mathbf{W} \Sigma \mathbf{W}^\top \mathbf{J}^\top. \quad (4)$$

For each pixel, the color and opacity contributed by each Gaussian are computed by evaluating Eq. (2) under its learned parameters. The blending of  $N$  ordered Gaussians overlapping a given pixel is defined as follows

$$\mathbf{C}_N = \sum_{n=1}^N \mathbf{c}_n \alpha_n \prod_{j=1}^{n-1} (1 - \alpha_j), \quad (5)$$

where  $\mathbf{c}_i$  and  $\alpha_i$  represent the color and opacity of the  $i$ -th Gaussian.

*Hologram Plane, Camera and Problem Definition.* We define the hologram plane as  $P \in \mathbb{C}^{n_x \times n_y}$ , where  $n_x, n_y \in \mathbb{Z}$ . The center of  $P$  is co-located with a camera pointing at the center of  $\Omega$  located at  $d$  distance away. We always position  $P$  based on the camera pose, location and angles,  $r = (x_c, y_c, z_c, \phi_x, \phi_y, \phi_z)$ , in Euler space. Each complex pixel  $P_i$  has a pixel pitch,  $\Delta x$ . Given input  $q = (r, \Delta x, n_x, n_y)$ , we compute the hologram plane by propagating the complex field rendered from Gaussian primitives to  $P(q)$

$$P(q) = U(q) * h, \quad (6)$$

where  $U(q)$  denotes the complex field rendered from the Gaussian primitives under camera parameters  $q$ , and  $h$  denotes spatial-domain convolutional kernel [Matsushima and Shimobaba 2009; Zheng et al. 2024]. Note that this propagation is an intermediate, differentiable rendering step. Our final goal is to optimize the Gaussian parameters such that the reconstructed intensities after inverse propagation match the focal-stack. A common propagation method is the band-limited ASM, where the transfer function  $H_z(f_x, f_y)$  encodes propagation by a distance  $z$  in the frequency domain and its inverse Fourier transform yields the spatial-domain kernel  $h$

$$H_z(f_x, f_y) = \begin{cases} \exp(j2\pi z \sqrt{\frac{1}{\lambda^2} - (f_x^2 + f_y^2)}), & \text{if } f_x^2 + f_y^2 \leq \frac{1}{\lambda^2} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

In this paper, we also utilize band-limited ASM as the propagation method for complex-valued holographic radiance fields modeling.

## 4 Method

In this section, we present our method using the terminology established in Sec. 3. Whereas classical radiance is a scalar energy measure  $W/(\text{sr} \cdot \text{m}^2)$  neglecting phase, our approach represents a complex-valued field, capturing both amplitude and phase information to model intensity, interference and diffraction. We use the term *complex-valued holographic radiance fields*, preserving conceptual links to prior 3D scene representations such as NeRF and 3DGS while reinterpreting radiance within the wave-optics framework. By embedding coherent light modeling into the established radiance field paradigm, our formulation attempts to bridge graphics and holographic rendering.

### 4.1 Complex-Valued Holographic Radiance Fields

We provide an overview of our method in Fig. 3. Given a camera pose,  $r$ , and a target hologram plane,  $P$ , we partition  $\Omega$  into multiple planes

in depth. At each depth plane, we render our learnable complex-valued 3D Gaussians into corresponding 2D projections. During projection, we utilize learnable probabilities for our 3D Gaussians that help us determine if a 3D Gaussian contributes to that selected depth plane. Note that learning these probabilities in terms of their contributions at each depth plane helps to inherently capture depth characteristics of a given scene. Once we determine the appropriate 2D projections for each layer, we propagate the complex field from each layer towards our camera to construct our  $P$ .

*Layer Construction.* For every 3D Gaussian  $\mathcal{G}_n$  with center  $\mathbf{x}_n$ , we project its 2D projections to plane  $\Pi_l$  based on learned assignment probabilities. Let  $\{\Pi_1, \Pi_2, \dots, \Pi_L\}$  be a set of discretized parallel planes in the volume,  $\Omega$ , positioned periodically along the optical axis in front of the view-dependent camera. Each plane  $\Pi_l$  is separated from its neighbors by a fixed depth interval of  $\Delta z$ . The planes span a volume of depth and the center of the volume is placed at a propagation distance,  $d$ , from the camera plane. In this arrangement, the entire depth is symmetrically distributed as in the common literature, ranging from:  $Z_1 = d - \frac{(L-1)}{2}\Delta z$  to  $Z_L = d + \frac{(L-1)}{2}\Delta z$ . Each plane  $\Pi_l$  is sampled on a 2D grid that matches the physical resolution of a SLM.

*Complex-Valued 3D Gaussian.* Consider a multi-plane case where Gaussians are projected onto  $L$  numbers of planes  $\{\Pi_l\}_{l=1}^L$ , we compute each 2D projection of 3D Gaussian following Eq. (4)

$$\mathcal{G}^{\text{proj}}(\mathbf{x}, \mathbf{R}, \mathbf{S}, \mathbf{J}, \mathbf{W}, u) = \exp\left(-\frac{1}{2}(u - \mu)^\top \Sigma'^{-1}(u - \mu)\right), \quad (8)$$

where  $u = (n_x, n_y)$  and  $\mu = \mathbf{J}\mathbf{W}\mathbf{x}$ . To model complex-valued holographic radiance fields, we have to modify the original definition of the 3D Gaussian primitive. **Specifically, each Gaussian primitive is now parameterized by**

$$k = (c_n, \mathbf{x}_n, \mathbf{R}_n, \mathbf{S}_n, \alpha_n, \underline{\boldsymbol{\varphi}}_n, \underline{\boldsymbol{\rho}}_n), \quad (9)$$

where we add two extra parameters:  $\boldsymbol{\varphi}_n \in \mathbb{R}^3$  to represent inherent phase values across wavelengths, and  $\boldsymbol{\rho}_n \in \mathbb{R}^L$  to represent the plane assignment probabilities, where each element lies in  $[0, 1]$ . Under this formulation,  $c_n$  denotes the inherent wave amplitude, replacing its original interpretation as color. The complex field of each 2D projection  $U_n$  then is given by

$$U_n = c_n \mathcal{G}_n^{\text{proj}} \exp(j\boldsymbol{\varphi}_n). \quad (10)$$

The wavenumber term is implicitly included in the phase parameter  $\boldsymbol{\varphi}_n$ , which represents the accumulated phase (including wavelength dependence) at the Gaussian level. When  $L = 1$ , plane assignment reduces to a single plane, the  $\boldsymbol{\rho}_n$  trivially equals 1.0, and the complex-valued holographic radiance field will generate 2D holograms. For the multi-plane case, we convert the raw, pre-activation logits  $\boldsymbol{\rho}'_n$  to actual probabilities via arg-max followed by one-hot encoding, resulting in a hard assignment vector  $\boldsymbol{\rho}_n$

$$\boldsymbol{\rho}_{n,l} = \text{OneHot}(\arg \max_l \boldsymbol{\rho}'_n) = \begin{cases} 1, & \text{if } l = \arg \max_l \boldsymbol{\rho}'_n, \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where  $\boldsymbol{\rho}_{n,l}$  is the assignment probability of the  $n$ -th Gaussian to the  $l$ -th depth plane. However, the argmax operation is non-differentiable, which impedes gradient-based training. To address this, we employ

the Straight-Through Estimator (STE) that allows gradient propagation through the discrete assignment

$$\begin{aligned} \text{Forward: } \boldsymbol{\rho}_n &= \text{OneHot}(\arg \max_l \boldsymbol{\rho}'_n), \\ \text{Backward: } \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}'_n} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}_n} \cdot \text{softmax}(\boldsymbol{\rho}'_n / \tau), \end{aligned} \quad (12)$$

where  $\tau$  is the temperature controlling the sharpness of the softmax approximation, in our case, we set  $\tau = 0.001$ , which makes the softmax function close to argmax operation. The STE is widely used in deep learning to approximate gradients through discrete operations [Yin et al. 2019], but it is not a standard component of neural rendering pipelines. We adopt it to our method to circumvent gradient propagation issues.

*Forward Recording and Inverse Propagation.* The final complex field at each plane  $\Pi_l$  is then defined as

$$U_{\Pi_l} = \sum_{n=1}^N \boldsymbol{\rho}_{n,l} U_n \alpha_n \prod_{j=1}^{n-1} (1 - \alpha_j) \boldsymbol{\rho}_{j,l}. \quad (13)$$

This formulation ensures each complex-valued 3D Gaussian contributes exclusively to its assigned plane. Here, the opacity  $\alpha_n$  modulates the amplitude of the emitted complex wavefront, representing electromagnetic field strength rather than color blending. It controls the wave emission intensity of each Gaussian and thereby indirectly shapes the resulting interference pattern, while the transmittance product  $\prod_{j=1}^{n-1} (1 - \alpha_j)$  accounts for wavefront attenuation due to occlusion by preceding Gaussians. The learned  $\boldsymbol{\rho}_n$  values guide assignment distribution across view-dependent plane configurations at different viewpoints, minimizing reconstruction loss during training. Unlike the alpha blending used in intensity-based approaches, which indiscriminately mixes RGB colors across all elements, our method creates plane-specific visibility by blending complex numbers (containing both amplitude and phase information) only among Gaussians assigned to the same depth plane to each pixel. Once each layer  $\Pi_l$  is populated by assigned 2D projections, we use the transfer function  $H$  to propagate  $\Pi_l$  towards the hologram plane  $P$ , a process we named *Forward Recording* in our framework. This wave-based propagation extends the traditional intensity neural rendering in Eq. (5). Specifically, for every  $\Pi_l$ , a  $(n_x^{(l)} \times n_y^{(l)})$  grid of complex samples, we compute

$$U_{\Pi_l \rightarrow P} = \mathcal{F}^{-1}\{H_{Z_l}(f_x, f_y) \cdot \mathcal{F}\{U_{\Pi_l}\}\}, \quad (14)$$

where  $Z_l$  is the distance between depth plane  $\Pi_l$  and the hologram plane  $P$ . The forward propagated field  $U_{\Pi_l \rightarrow P}$  is then added with the contributions from the other planes to record the final complex 3D hologram [Shimobaba et al. 2009],

$$P = \sum_{l=1}^L U_{\Pi_l \rightarrow P}. \quad (15)$$

Since each plane is a 2D raster, the computation cost scales linearly with the number of layers  $L$  and is dominated by FFTs of size  $n_x^{(l)} \times n_y^{(l)}$ . After obtaining  $P$ , we propagate it back to each depth plane. The back-propagated complex field  $U_{P \rightarrow \Pi_l}$  at depth plane  $\Pi_l$  is then computed as

$$U_{P \rightarrow \Pi_l} = \mathcal{F}^{-1}\{H_{-Z_l}(f_x, f_y) \cdot \mathcal{F}\{P\}\}, \quad (16)$$

where we obtain its intensity  $I_l = |U_{P \rightarrow \Pi_l}|^2$  as the final rendered image, we name this process *Inverse Propagation* in our framework. Naively, we can supervise the reconstructed intensities  $I$  from back-propagated complex field  $U$  against the ground-truth image  $I_{gt}$  per depth plane  $l$

$$\mathcal{L}_{MSE} = \frac{1}{L} \sum_{l=1}^L \|I_l - I_{gt,l}\|^2. \quad (17)$$

The supervision consists of focal-stack intensity images rendered from captured or synthetic multi-view data [Kavaklı et al. 2023a], serving as ground-truth observations and are not produced by any learned baseline. To further improve the image quality of the defocus region, we utilize two complementary loss functions. First, the reconstruction loss  $\mathcal{L}_{recon}$  introduced by Kavaklı et al. [2023a], computed as

$$\begin{aligned} \mathcal{L}_{recon} = & \frac{1}{L} \sum_{l=1}^L (\|I_l - I_{gt,l}\|^2 \\ & + \|I_l \cdot M_l - I_{gt,l} \cdot M_l\|^2 + \|I_l \cdot I_{gt,l} - I_{gt,l} \cdot I_{gt,l}\|^2), \end{aligned} \quad (18)$$

where  $M_l$  is the binary mask for depth plane  $\Pi_l$  generated from the target image and its quantized depth. Additionally, we employ the SSIM loss  $\mathcal{L}_{SSIM}$  defined as

$$\mathcal{L}_{SSIM} = \frac{1}{L} \sum_{l=1}^L \lambda_1 \cdot (1 - \text{SSIM}(I_l, I_{gt,l})), \quad (19)$$

where  $\lambda_1 = 0.005$ . The final training loss is  $\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{SSIM}$ . Neither the complex hologram values nor the plane-assignment probabilities are directly supervised in our method, as no GT hologram or plane label per Gaussian is available. Both variables are learned implicitly through reconstruction loss on focal-stack intensities.

*View-Dependent Plane Configuration.* At each viewpoint, the depth planes  $\{\Pi_l\}_{l=1}^L$  are always positioned parallel to the hologram plane  $P$ , ensuring standard band-limited ASM remains valid. To our knowledge, there is not a work adopting off-axis beam propagation in this context, including the prior work [Choi et al. 2025]. For novel views with camera pose  $r'$ , we recompute projection matrices  $\mathbf{J}'$  and  $\mathbf{W}'$  to project Gaussians onto the reconfigured planes. The plane assignment probabilities  $\rho_n$  are learned across training views, serving as computational latent variables that guide Gaussian distribution across depth planes.

## 4.2 Fast Differentiable Complex-Valued Rasterizer

Building upon the tile-based architecture from 3DGS, we extend the system to support complex field calculations for holographic rendering while maintaining fast sorting and efficient parallel processing. Our method preserves the core  $16 \times 16$  tile structure, computing screen-space extents with additional consideration for phase-dependent effects during preprocessing.

For the forward pass, we adapt tile-based rendering to track both real and imaginary components of Gaussians. Each thread block collaboratively loads Gaussian data into shared memory, with threads processing individual pixels by accumulating complex-valued field contributions using fast trigonometric operations. Multi-plane rendering is handled through plane-specific filtering directly in the

kernel, skipping Gaussians with low assignment probabilities to minimize thread divergence while supporting depth-dependent complex field accumulation. Per pixel, we track final transmittance and the last contributing Gaussian’s position per plane for correct backward gradients.

The backward pass maintains the same tile-based structure but traverses Gaussians back-to-front, reusing the sorted array from the forward pass with pixels processing only up to their recorded last contributor. This ensures constant memory overhead regardless of scene complexity. We store only the final accumulated opacity per pixel during forward pass, then recover intermediate opacity values during the backward pass by dividing the final opacity by each Gaussian’s contribution, eliminating the need for long opacity lists while enabling accurate gradient computation. This implementation efficiently renders complex-valued holographic radiance fields while retaining the performance benefits of the original tile-based rendering structure. For details of the forward operation and backward gradient computation, please refer to Supplementary Sec. D.

## 4.3 scene geometry-aware Amplitude and Phase Representations

*Existing CGH Methods.* A limitation in existing CGH methods is their dependency on optimizing the complex field for a fixed geometric relationship between the scene geometry and the hologram plane. In these methods, a hologram plane  $P_0(x, y)$  is designed to encode the entire 3D scene when viewed from a specific viewpoint, with the complex field at the viewing plane described as:

$$U_0(x', y') = \mathcal{F}^{-1}\{H_z(f_x, f_y) \cdot \mathcal{F}\{P_0(x, y) \cdot R(x, y)\}\}, \quad (20)$$

where  $R(x, y)$  is the reference wave. While a properly estimated 3D hologram inherently contains information for different viewing angles within the Space-Bandwidth Product (SBP) limits of the pixel pitch—functioning as a “window” into a virtual world that observers can view from different positions—the issue arises when the *geometric relationship between the scene geometry and the hologram plane changes*. Unlike assuming a stationary or narrowly perturbed observer at the center of projection, any significant movement of the camera plane itself within the 3D world demands recalculating the hologram:

$$P_\theta(x, y) \neq P_0(x, y), \quad (21)$$

where  $\theta$  represents the new viewpoint when the geometric relationship between scene geometry and hologram plane changes. Existing CGH methods estimate, optimize, or calculate closed-form solutions of  $P(x, y)$  to produce desired intensity patterns, treating each viewpoint as an independent optimization problem. As shown in Fig. 2, when the scene-to-hologram relationship changes, the previously computed hologram leads to visuals that are highly distorted, resembling noise. Although neural networks enable efficient recalculation of holograms for new configurations, these learned methods fail to maintain consistent amplitude and phase across different viewpoints. Instead, neural networks essentially create a new, unrelated data-driven solution, which lacks interpretability and physical consistency; in real-world scenarios, the amplitude and phase relationships from an object remain invariant regardless of its geometric relationship to the observation plane. In traditional intensity-based

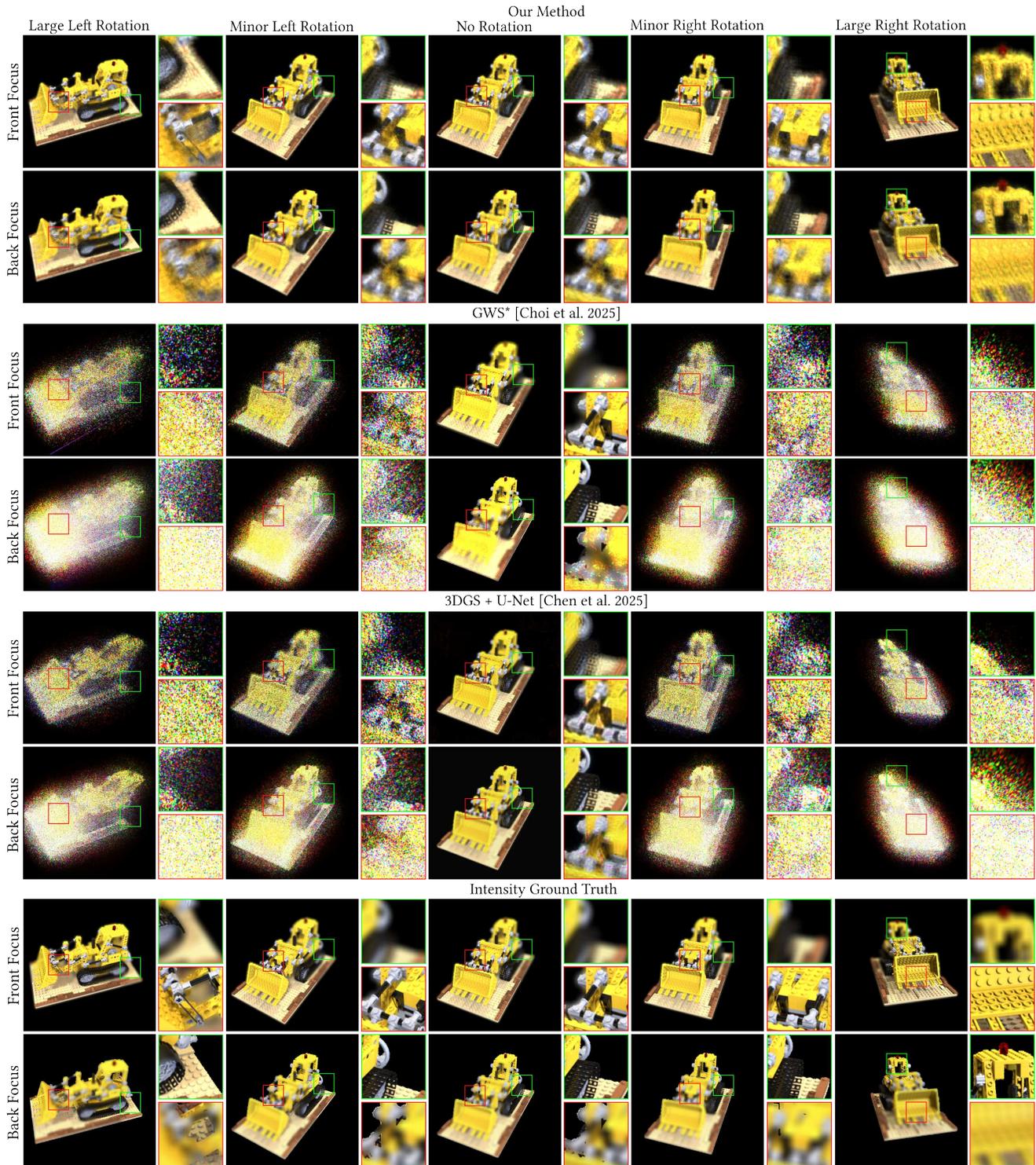


Fig. 4. Comparison of different hologram synthesis methods across different viewpoints in simulation. The top row shows our method with scene geometry-aware representations from large left to right rotations for novel views. The middle rows show the existing CGH methods that rely on intensity-based intermediaries, which fail to maintain consistency across novel views. The bottom row shows the intensity ground truth. \* We reimplement [Choi et al. 2025] to demonstrate the results of GWS; The original non-rotational result shows higher image quality and different defocus blur than our reimplement.

rasterization, each pixel depends only on a handful of nearby fragments; in CGH, by contrast, every primitive contributes across a large “sub-hologram” footprint on the full resolution hologram, and one must accumulate complex-valued wavefronts from all primitives at all spatial frequencies. Primitive-based methods that maintain physical accuracy (e.g., GWS [Choi et al. 2025] or polygon-based CGH) require proper handling of occlusions and interference effects through complex wavefront accumulation:

$$P(x, y) = \sum_{n=1}^N \mathcal{F}^{-1}\{H_{z_n}(f_x, f_y) \cdot \mathcal{F}\{\mathcal{G}_n(x, y) \cdot \mathcal{T}_n(x, y)\}\}, \quad (22)$$

where  $\mathcal{T}_n(x, y)$  represents the accumulated transmittance for occlusion of the  $n$ -th primitive. This formulation requires expensive convolution for each primitive across the entire hologram plane, yielding complexity of  $O(N_{\text{primitives}} \times N_{\text{freq}} \log N_{\text{freq}})$ , where  $N_{\text{freq}}$  is the total number of frequency bins. To improve efficiency, occlusion effects can be neglected by approximating  $\mathcal{T}_n(x, y) \approx 1$ :

$$P(x, y) \approx \sum_{n=1}^N \mathcal{F}^{-1}\{H_{z_n}(f_x, f_y) \cdot \mathcal{F}\{\mathcal{G}_n(x, y)\}\}. \quad (23)$$

While this reduces complexity to  $O(N_{\text{primitives}} \times N_{\text{freq}})$  and enables better parallelization, it sacrifices accurate occlusion modeling and still requires substantial computation, as each primitive must be re-evaluated across the entire hologram plane. Existing CGH methods follow an Eulerian computational paradigm, where the hologram plane coordinates remain fixed while complex fields are recalculated for each viewpoint.

*Our Method.* In contrast, our approach differs fundamentally by modeling the complex field directly in 3D space using Gaussian primitives with intrinsic phase properties — a Lagrangian perspective where the content remains fixed while the viewpoint changes. Each complex-valued Gaussian  $\mathcal{G}_n$  has an explicitly defined phase parameter  $\varphi_n$  that represents its inherent holographic properties, independent to camera parameters. We emphasize that this per-Gaussian phase parameter does not correspond to the absolute optical phase measured at the sensor in real life and does not change with the position of the observer or the camera. Instead, it serves as a learned intrinsic reference phase, characterizing the local wave emission of each Gaussian within the scene’s Lagrangian coordinate. View-dependent and position-dependent interference will arise subsequently through light propagation. Although the intrinsic phase parameters remain fixed across viewpoints, the result of interference observed at the camera plane will vary consistently with scene geometry. Therefore, the complex field contribution of each Gaussian is uniquely determined by its 3D position, orientation, and intrinsic phase, remaining coherent across views.

This Lagrangian formulation enables us to avoid recalculations in the Eulerian approach, avoids the expensive per-frequency summation, and instead pushes all of the work into two highly optimized GPU-friendly stages: 1. *Tile-based Complex-Valued Rasterization*: As described in Sec. 4.2, by dividing the screen into  $16 \times 16$  tiles, we extend the 3DGS rasterizer to accumulate complex-valued Gaussians directly in image space and reduce the rasterization cost to  $O(N_{\text{primitives}})$ . 2. *FFT-based layer propagation*: Instead of summing

each primitive’s contribution at every spatial frequency, we collect all Gaussians onto  $L$  depth planes once, and then propagate these planes to the hologram plane using 2D FFTs. This reduces the propagation cost to  $O(L \times N_{\text{res}} \log N_{\text{res}})$ , where  $N_{\text{res}} = n_x \times n_y$  is the number of pixels per depth plane, independent to the number of primitives. In this case, since  $L$  does not grow with  $N_{\text{primitives}}$  and  $L \times N \log N$  is much smaller than  $N_{\text{primitives}} \times N_{\text{freq}}$  when the number of primitives is large, our approach achieves inherently better scaling. This results in a more straightforward holographic rendering pipeline that maintains constant propagation cost regardless of scene complexity. Mathematically, consider two different viewpoints with respective projection matrices  $\mathbf{J}_1, \mathbf{W}_1$  and  $\mathbf{J}_2, \mathbf{W}_2$ . For a given  $\mathcal{G}_n$ , while its projected complex field will differ between viewpoints, the underlying phase relationship remains consistent:

$$\begin{aligned} U_n^{(1)} &= \mathbf{c}_n \mathcal{G}_n^{\text{proj}}(\mathbf{J}_1, \mathbf{W}_1) \exp(j\varphi_n) \\ U_n^{(2)} &= \mathbf{c}_n \mathcal{G}_n^{\text{proj}}(\mathbf{J}_2, \mathbf{W}_2) \exp(j\varphi_n). \end{aligned} \quad (24)$$

This phase consistency is further maintained through the *Forward Recording* process:

$$P^{(i)} = \sum_{l=1}^L \mathcal{F}^{-1}\{H_{Z_l}(f_x, f_y) \cdot \mathcal{F}\{U_{\Pi_l}^{(i)}\}\}, \quad (25)$$

where  $U_{\Pi_l}^{(i)}$  represents the complex field at depth plane  $\Pi_l$  for viewpoint  $i$ , and  $P^{(i)}$  is the recorded hologram. As shown in Fig. 4, this consistency becomes apparent when examining the reconstructed wavefront across different viewpoints. In real life, light maintains intrinsic phase offsets at emission that are tied to scene geometry, while the observed phase relationships at the sensor vary with viewpoint through wave propagation and superposition [Jang et al. 2024]. Complex-valued holographic radiance field attempts to mimic this physical behavior, ensuring that the amplitude and phase differences between scene elements remain invariant, leading to physically-consistent interference and diffraction effects. This represents a paradigm shift—from treating the amplitude and phase as computational variables applied after intensity determination on the hologram plane, to modeling them as an intrinsic and meaningful property of the 3D scene itself. By combining the Lagrangian perspective of 3DGS with holography, we achieve modeling and rendering of a complex-valued holographic radiance field, eliminating the need for expensive per-configuration, per-primitive wave calculations.

## 5 Implementation

We employ the ground-truth images and rendered depth information to generate target reconstruction images using the multiplane generation pipeline by Kavaklı et al. [2023b]. Additionally, we employ the fused SSIM proposed by [Mallick et al. 2024] to speed up the training process. For densification, we adopt the original strategy from 3DGS, with the exceptions that the densifying frequency is 300 steps and the opacity resetting is replaced by the regularization method proposed by Rota Bulò et al. [2024].

We choose Adan [Xie et al. 2024] as the optimizer ( $\beta_1 = 0.9, \beta_2 = 0.99$ ) and train our complex-valued Gaussians for 20000 steps using parameter-specific learning rates: 0.005 for scales, 0.0025 for both

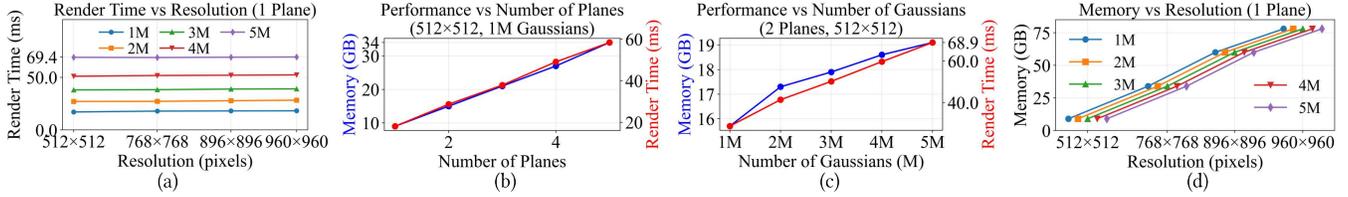


Fig. 5. Performance and memory usage analysis. (a) Render time vs resolution for different Gaussian counts under 1 plane. (b) Memory usage & render time vs number of depth planes. (c) Memory usage & render time vs number of Gaussians. (d) Memory usage vs resolution for different numbers of Gaussians.

amplitude and phase, 0.025 for opacity, 0.001 for rotation, and a base learning rate of 0.01 for means and plane assignment probabilities. Our choice of optimizer follows [Zhang et al. 2024], while the learning rate is adopted from the original 3DGS implementation. We apply CosineAnnealingLR [Loshchilov and Hutter 2017] only to the learning rates of means and plane assignment probabilities (minimum 0.00001) for smoother convergence, while keeping the others fixed throughout training. For holographic configuration, we follow the most recent literature [Akşit and Itoh 2023; Shi et al. 2021, 2022] and choose a propagation distance of 2 mm, volume depth of 4 mm, pixel pitch of 3.74  $\mu\text{m}$ , and wavelengths of 639, 532, and 473 nm. All experiments are conducted on a NVIDIA A100 80G GPU. For details on dataset-specific camera viewpoint sampling strategies, please refer to Supplementary Sec. C.

## 6 Evaluation

We evaluate our method on three standard datasets, including NeRF Synthetic [Mildenhall et al. 2021], LLFF [Mildenhall et al. 2019], and Mip-NeRF 360 [Barron et al. 2022].

### 6.1 Baseline Selection

We compare our method against a two-stage baseline that generates holograms using a U-Net [Ronneberger et al. 2015] from RGBD images rendered from a pretrained intensity-based 3DGS scene. The baseline is selected for several reasons. First, although neural network estimates holograms that are not scene-geometric aware, it achieves millisecond-level inference, substantially faster than optimization-based CGH methods. Second, despite introducing unnatural defocus effects and a limited eye box, the network’s denoising capability suppresses speckle noise and artifacts commonly found in the other CGH methods, yielding the highest quantitative image quality. As such, this method represents a theoretical upper bound on achievable image quality under ideal conditions. We therefore treat 3DGS + U-Net (GT) as the *Image Quality Ground Truth* for benchmarking. In addition, we evaluate a *Viewpoint Relationship Variation* setting, denoted 3DGS + U-Net (Vary), where holograms are generated without recalculation when the relationship between scene geometry and viewpoint changes.

From a computational perspective, 3DGS + U-Net requires rendering RGBD images from the pretrained 3DGS scene at each queried viewpoint, followed by neural inference to generate the hologram. Similarly, GWS first renders the 2DGS scene, then computes closed-form wave propagation for each Gaussian primitive at the queried

viewpoint. Both adopt an Eulerian paradigm, recalculating holograms for every novel view at camera plane. We use 3DGS + U-Net as the primary baseline because the source code of GWS is currently unavailable; our reimplementations yields lower image quality, and its reconstructions exhibit different defocus blur characteristics than the official implementation, making the comparison inaccurate and unfair.

### 6.2 Inference Time

Table 3 demonstrates our method’s efficiency. Compared with GWS [Choi et al. 2025], our method is inherently more scalable, achieving 30x-10,000x speedup while maintaining view consistency. While 3DGS + U-Net [Chen et al. 2025] offers faster rendering, it requires recalculation of the hologram per viewpoint, which has no scene geometry-awareness. Even with a large number of Gaussians (5M), our method maintains reasonable performance (69 ms), achieving a balance between computational efficiency and maintaining scene geometry-aware representations. Fig. 5 illustrates the scalability of our method; the render time remains relatively constant across resolutions due to tile-based rasterization, while both memory usage and render time scale linearly with the number of depth planes and Gaussian primitives, demonstrating the computational efficiency of our method compared to traditional CGH methods that rely on per-primitive diffraction calculation. For ablation study of the computational contribution of tile-based rasterization and FFT-based propagation, please refer to Supplementary Sec. E.

### 6.3 Complex Field Discontinuities: Beyond Intensity-Based Smooth Interpolation

**6.3.1 Quantitative Analysis.** Table 2 and 4 show the training images performance of our method and 3DGS + U-Net on the NeRF Synthetic, LLFF, and Mip-NeRF 360 datasets, respectively. We evaluate the image quality using PSNR, SSIM, and LPIPS [2018a].

The *Viewpoint Relationship Variation* results show that 3DGS + U-Net (Vary) fails to generate correct holograms when viewpoint changes without recalculation, with PSNR dropping by 21.5 dB, 20.4 dB, and 19.6 dB on NeRF Synthetic, LLFF, and Mip-NeRF 360 datasets, respectively, and SSIM decreases by over 0.74 across all 17 scenes. In contrast, our method maintains robust performance across novel views, achieving 18.7 dB, 18.5 dB, and 13.9 dB PSNR improvements over 3DGS + U-Net (Vary).

For the *Image Quality Ground Truth*, our method achieves promising performance on the NeRF Synthetic and LLFF datasets, with PSNR gaps of 2.8 dB and 1.9 dB and SSIM differences of 0.05 and

Table 2. Quality comparison on NeRF Synthetic and LLFF. 3DGS + U-Net (GT) refers to the *Image Quality Ground Truth* that requires recalculation of hologram per viewpoint. 3DGS + U-Net (Vary) represents the *Viewpoint Relationship Variation* without recalculation. Metrics are reported as PSNR/SSIM/LPIPS, respectively. Bold indicates best performance excluding GT baseline. For more information about baselines, please refer to Sec. 6.1.

Method	NeRF Synthetic (Test Resolution 800×800)								Mean
	chair	drums	figus	hotdog	lego	materials	mic	ship	
3DGS + U-Net (GT)	29.3/0.91/0.13	28.9/0.90/0.14	31.3/0.95/0.10	30.4/0.94/0.15	28.1/0.89/0.15	29.1/0.92/0.14	32.7/0.95/0.10	29.1/0.90/0.18	29.9/0.92/0.14
3DGS + U-Net (Vary)	7.2/0.15/0.71	9.8/0.12/0.77	8.1/0.18/0.72	8.5/0.14/0.85	7.9/0.09/0.81	7.8/0.16/0.75	8.4/0.19/0.79	9.3/0.11/0.79	8.4/0.14/0.77
Our Method	<b>27.6/0.89/0.14</b>	<b>26.1/0.86/0.17</b>	<b>28.0/0.91/0.13</b>	<b>29.1/0.90/0.16</b>	<b>24.3/0.83/0.21</b>	<b>28.3/0.88/0.17</b>	<b>29.2/0.91/0.14</b>	<b>24.9/0.81/0.23</b>	<b>27.1/0.87/0.17</b>

Method	LLFF (Test Resolution 960×640)						Mean	
	fern	flower	fortress	horns	orchids	room		trex
3DGS + U-Net (GT)	29.6/0.90/0.35	29.4/0.89/0.32	27.4/0.88/0.37	28.8/0.91/0.37	27.5/0.84/0.39	28.9/0.93/0.36	26.1/0.83/0.41	28.3/0.88/0.37
3DGS + U-Net (Vary)	8.3/0.17/0.78	7.1/0.13/0.82	8.8/0.11/0.85	9.2/0.19/0.79	5.7/0.12/0.83	8.9/0.16/0.80	7.4/0.12/0.86	7.9/0.14/0.82
Our Method	<b>27.9/0.78/0.39</b>	<b>28.1/0.80/0.37</b>	<b>25.2/0.73/0.45</b>	<b>26.7/0.79/0.43</b>	<b>25.1/0.73/0.46</b>	<b>27.3/0.81/0.41</b>	<b>24.5/0.75/0.45</b>	<b>26.4/0.77/0.42</b>

Table 3. Comparison of inference times across different Gaussian primitive based hologram synthesis methods at 800×800 resolution. Inference time for 3DGS + U-Net includes per-view recomputation, consisting of 3DGS rendering and network inference for each queried viewpoint.

Method	Number of $\mathcal{G}$ s	Inference Time
Our Method	200K	10 ms
	5M	69 ms
3DGS + U-Net [Chen et al. 2025]	200K	8 ms
	5M	29 ms
GWS (Fast) [Choi et al. 2025]	15K	> 3 s
	200K	> 40 s
	5M	> 15 min
GWS (Exact) [Choi et al. 2025]	15K	> 1 min
	200K	> 13 min
	5M	> 5 hrs

0.11, respectively, compared to 3DGS + U-Net (GT). However, performance degradation of our method becomes significant on Mip-NeRF 360 dataset with 5.7 dB PSNR drop and 0.42 SSIM decrease. This progressive degradation from controlled synthetic scenes to complex in-the-wild environments stems from the fundamental challenge of representing spatially discontinuous complex fields to model interference and diffraction. For more analysis of gaussian distribution and statistics of holograms rendered by our method, please refer to Supplementary Sec. F.

**6.3.2 Complex Field Discontinuities.** In intensity-based radiance fields, the goal is to model view-dependent appearance through opacity and color. This representation handles sparse viewpoint sampling effectively because intensity variations generally follow smooth, low-frequency patterns that can be interpolated between viewpoints. However, modeling complex-valued holographic radiance fields introduce a challenge related to motion parallax. For a camera translation, motion parallax causes objects at different depths to experience different amounts of apparent displacement. Objects at depths  $d_1$  and  $d_2$  exhibit displacements of:

$$\Delta s_1 = \frac{f \cdot \Delta t}{d_1}, \quad \Delta s_2 = \frac{f \cdot \Delta t}{d_2}, \quad (26)$$

where  $f$  is the focal length and  $\Delta t$  is the camera translation. When  $d_2 \gg d_1$ , the motion parallax difference  $\Delta s_1 - \Delta s_2$  becomes significant. This motion parallax translates directly into phase differences in complex-valued holographic radiance fields. Between consecutive camera viewpoints, these Gaussian primitives will accumulate different phase shifts:

$$\Delta \phi_1 = \frac{2\pi}{\lambda} \Delta s_1, \quad \Delta \phi_2 = \frac{2\pi}{\lambda} \Delta s_2. \quad (27)$$

The issue arises when adjacent pixels correspond to objects with vastly different depths, creating sharp spatial discontinuities in phase distribution. Consider two Gaussian primitives at depths  $d_1$  and  $d_2$  that are spatially adjacent in the hologram plane, the contrasting phase evolution during camera motion will introduce visual artifacts. For adjacent pixels separated by distance  $\Delta x$ , the spatial phase gradient is defined as:

$$\frac{\partial \phi}{\partial x} = \frac{\Delta \phi_2 - \Delta \phi_1}{\Delta x} = \frac{2\pi}{\lambda} \frac{f \cdot \Delta t \cdot (d_2 - d_1)}{d_1 d_2 \Delta x}. \quad (28)$$

When the scene contains large depth discontinuities ( $d_2 \gg d_1$ ), the spatial phase gradient will become very large. This creates interpolation challenges absent in intensity-based representations. Unlike intensity values, which are continuous and monotonic, phase values wrap at  $2\pi$  boundaries. Large spatial phase gradients cause adjacent pixels to differ by multiple cycles, creating ambiguity in interpolation paths that intensity-based representations do not encounter. To evaluate our method against complex field discontinuities, we carefully choose and analyze challenging scenes with significant depth variations from commonly used datasets that exhibit different motion parallax characteristics.

As shown in Fig. 6, the NeRF Synthetic dataset benefits from controlled capture conditions with relatively uniform depth distributions within each scene. In contrast, LLFF and Mip-NeRF 360 datasets contain in-the-wild scenes with larger depth variations. These scenes create sharp phase boundaries where the complex field changes abruptly in between viewpoints, breaking the spatial coherence assumptions of intensity-based representations. While the 3DGS + U-Net method maintains visual quality by first modeling geometry and appearance through intensity-based 3DGS (which handles motion parallax effectively) and then estimating hologram

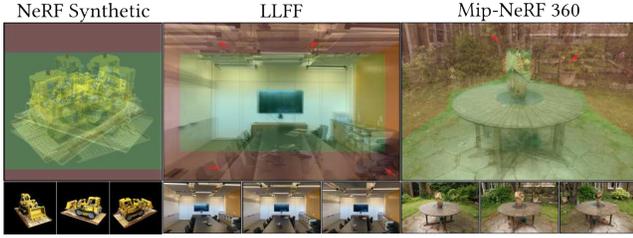


Fig. 6. Motion parallax across datasets: NeRF Synthetic (left) shows a controlled scene with minimal parallax; LLFF (center) shows an indoor scene with vast-moving foreground motion (red region); Mip-NeRF 360 (right) shows an in-the-wild scene with wide background shifts (red region). Regions exhibiting moderate motion parallax that does not cause phase discontinuities in our experiments are marked in green.

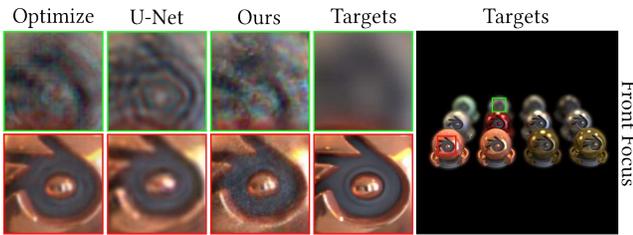


Fig. 7. Defocus blur comparison in holographic reconstructions between optimization [Kavaklı et al. 2023b], 3DGS + U-Net [Chen et al. 2025], and our method against target images.

as a secondary task, this two-stage CGH approach sacrifices the scene geometry-awareness across different viewpoints. This limitation highlights the trade-off between the spatial coherence requirements of intensity-based representations and the phase discontinuities introduced by motion parallax in complex-valued holographic radiance fields.

Table 4. Quality comparison on Mip-NeRF 360. 3DGS + U-Net (GT) refers to the *Image Quality Ground Truth* that requires recalculation of hologram per viewpoint. 3DGS + U-Net (Vary) represents the *Viewpoint Relationship Variation* without recalculation. Metrics are reported as PSNR/SSIM/LPIPS, respectively.

Method	Mip-NeRF 360 (Test Resolution 960×640)	
	garden	kitchen
3DGS + U-Net (GT)	25.9/0.87/0.35	28.1/0.91/0.34
3DGS + U-Net (Vary)	7.9/0.11/0.72	6.9/0.13/0.78
Our Method	20.2/0.45/0.61	22.4/0.49/0.53

## 6.4 Qualitative Analysis

*Defocus Blur Comparison.* Fig. 7 compares simulated defocus blur across three CGH methods: optimization, 3DGS + U-Net, and ours. Our method generates perceptually more plausible defocus blur, similar to the optimization-based result without per-view recalculation. 3DGS + U-Net’s result suffers from a structured, fringing effect in defocus blur, which is commonly found in learned CGH methods [Chen et al. 2025; Shi et al. 2021].

*Novel View Image Quality Analysis.* We evaluate novel view image quality on the NeRF Synthetic, LLFF, and Mip-NeRF 360 datasets—shown in Fig. 8, Fig. 9, and Fig. 10, respectively—using both simulation and experimental captures. For the NeRF Synthetic dataset, our method achieves visually consistent results across different datasets and on-par image quality compared to the baseline.

For the LLFF dataset, our method preserves image quality effectively in regions with stable depth content, particularly for background elements. However, artifacts become apparent in areas containing near-field objects that exhibit rapid motion across viewpoints, such as foreground furniture and close-range structural elements. These artifacts manifest as phase inconsistencies and visual distortions, primarily in regions where objects undergo significant displacement due to motion parallax.

For the Mip-NeRF 360 dataset, the most significant quality loss of our method occurs in distant background regions that experience substantial motion across large viewpoint changes, such as far-field vegetation and architectural structures. Objects at intermediate and near depths maintain reasonable reconstruction quality. This depth-dependent performance indicates that limitations arise from phase discontinuities caused by motion parallax rather than deficiencies in our complex-valued representations, as evidenced by the preserved quality in spatially coherent depth regions. For more simulated and captured results, please refer to Supplementary Sec. G.

*Phase-only Hologram Conversion.* To display our complex-valued holograms on a phase-only SLM, we leverage a propagation-based optimization approach that converts amplitude-phase into phase-only representations. Given our complex hologram  $P = Ae^{j\varphi}$  with amplitude  $A$  and phase  $\varphi$ , we aim to obtain a corresponding phase-only hologram  $P_\varphi = e^{j\varphi_{\text{opt}}}$  where the amplitude is unity and the phase  $\varphi_{\text{opt}}$  is optimized. We propagate both holograms to the same set of depth planes and minimize their propagation differences. For each depth plane  $\Pi_l$ , we compute the propagated complex fields

$$U_l = \mathcal{F}^{-1}\{H_{-Z_l}(f_x, f_y) \cdot \mathcal{F}\{P\}\}, \quad (29)$$

$$U_{\varphi,l} = \mathcal{F}^{-1}\{H_{-Z_l}(f_x, f_y) \cdot \mathcal{F}\{P_\varphi\}\},$$

and the optimization minimizes the difference between the real and imaginary components across all depth planes

$$\mathcal{L}_\varphi = \frac{1}{L} \sum_{l=1}^L \|U_l - U_{\varphi,l}\|^2 + \mathcal{L}_{SSIM}(U_l, U_{\varphi,l}). \quad (30)$$

Future variants could adopt learned approaches, such as U-Net and others, to replace optimization and enable more efficient conversion.

## 7 Discussion and Future Work

### 7.1 Limitations

*Wave Propagation Overhead.* Although our method renders 3D holograms effectively, the rendering process involves splatting Gaussian primitives onto multiple depth planes, followed by wave propagation using the band-limited ASM. The band-limited ASM method is not as efficient as differential splatting, particularly at high resolutions, as it involves multiple high-resolution FFTs to compute accurate wave propagation, which inevitably increases the computation and the memory cost. A more efficient light propagation

method [Zhan et al. 2025] is required to further reduce the memory cost and match the rendering speed of intensity-based 3DGS.

*Motion Parallax.* Our method’s reliance on Gaussian primitives for smooth spatial interpolation creates limitations in scenes with significant motion parallax. Objects at dramatically different depths exhibit varying displacements during viewpoint changes, generating sharp phase discontinuities that violate the smoothness assumptions of intensity-based representations. This explains our degraded results on in-the-wild datasets with large depth variations. Future work could explore hierarchical Gaussians [Wang et al. 2025] to adaptively model phase discontinuities at multiple scales.

*Incoherent Illumination.* Our method operates under the assumption of fully coherent illumination, where wavefronts exhibit stable phase relationships and produce predictable interference patterns. In contrast, most natural light sources are only partially coherent or incoherent, with phase correlations that degrade over time or across spatial extents. Our model’s reliance on coherent light leads to sensitivity to depth-induced phase artifacts, and partial coherence may help mitigate these discontinuities, highlighting an important direction for future work.

*Learned Plane Assignment.* The plane assignment probabilities in our method are a computational latent variable rather than a physically-accurate depth labeling. The consistency across interpolated novel viewpoints is learned implicitly from the training view distribution, rather than guaranteed by a physical model. While effective in practice, this approach is a pragmatic compromise. Future work should explore more elegant and physically grounded alternatives.

## 7.2 Future Work

Our work opens the door to potential advances in holographic rendering technology.

*Scene Relighting.* By directly modeling the complex-valued holographic radiance fields, a future variant of our method could support scene relighting [Guo et al. 2019]. Building on prior research in reflectance displays [Glasner et al. 2014], our approach could extend spatially varying reflectance functions through learned 3D structure and phase information that responds to illumination changes passively and reactively.

*Pupil-aware Holography.* A future variant could explore estimating complex-valued holographic radiance fields that dynamically adapt to pupil movements under novel views [Chakravarthula et al. 2022]. Moreover, our method inherently contains light field information within the complex-valued 3D representations. With appropriate training objectives, a future extension of our approach could directly support real-time 4D CGH rendering [Kim et al. 2024] without requiring explicit light field extraction.

*Holographic Camera.* Another promising direction lies in bridging synthetic complex-valued holographic radiance fields with real-world holographic acquisition. Recent advances in incoherent holographic cameras [Li et al. 2025; Yu et al. 2023] can capture both intensity and phase information from real 3D scenes under natural

illumination without coherent laser sources. Integrating such technology with our framework could enable modeling the true wave optics characteristics of real objects, moving beyond current RGB reproduction to authentic physical phase properties.

*Physically-informed Color Representation.* In the physical world, light amplitude is naturally characterized by its spectral distribution, not through mathematical constructs like Spherical Harmonic (SH). To maintain scene geometry-aware representations, our complex-valued holographic radiance field uses only RGB triplets to define the intrinsic amplitude of Gaussians to align with the intrinsic phase values. Despite this simplification, our experiments demonstrate that complex-valued holographic radiance field preserves view-dependent reflectance properties of 3D scenes without requiring high-dimensional view-dependent features. Compared to SH, which typically contains 48 parameters, our method requires only 6 (3 for amplitude, 3 for phase), resulting in reduced model size and memory footprint. Future work could examine the theoretical basis of this observation and assess whether physically plausible parameterizations further enhance radiance field expressiveness.

*Conclusion.* Our work introduces a novel approach to support holographic representations in Gaussian Splatting by utilizing our novel 3D complex-valued Gaussians as primitives. We demonstrate that this method leads to a coherent novel view synthesis that is faithful to the geometry of a 3D scene in terms of both amplitude and phase. These findings from our work demonstrate the potential for more physically-accurate 3D scene representations, helping holograms to replace 2D images and holographic displays to be compatible and practical as 3D screens in the future.

## References

- Lukas Ahrenberg, Philip Benzie, Marcus Magnor, and John Watson. 2008. Computer generated holograms from three dimensional meshes using an analytic light transport model. *Applied optics* 47, 10 (2008), 1567–1574.
- Kaan Akşit and Yuta Itoh. 2023. HoloBeam: Paper-Thin Near-Eye Displays. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 581–591.
- Jungkwuen An, Kanghee Won, Young Kim, Jong-Young Hong, Hojung Kim, Yongkyu Kim, Hoon Song, Chilsung Choi, Yunhee Kim, Juwon Seo, et al. 2020. Slim-panel holographic video display. *Nature communications* 11, 1 (2020), 5568.
- Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5855–5864.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5470–5479.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19697–19705.
- Praneeth Chakravarthula, Seung-Hwan Baek, Florian Schiffrers, Ethan Tseng, Grace Kuo, Andrew Maimone, Nathan Matsuda, Oliver Cossairt, Douglas Lanman, and Felix Heide. 2022. Pupil-Aware Holography. *ACM Trans. Graph.* 41, 6, Article 212 (Nov. 2022), 15 pages. <https://doi.org/10.1145/3550454.3555508>
- Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM transactions on graphics (TOG)* 32, 3 (2013), 1–12.
- Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensorRT: Tensorial radiance fields. In *European conference on computer vision*. Springer, 333–350.
- Kenneth Chen, Anzhou Wen, Yunxiang Zhang, Praneeth Chakravarthula, and Qi Sun. 2025. View synthesis for 3D computer-generated holograms using deep neural fields. *Optics Express* 33, 9 (2025), 19399–19408.
- Suyeon Choi, Brian Chao, Jacqueline Yang, Manu Gopakumar, and Gordon Wetzstein. 2025. Gaussian Wave Splatting for Computer-Generated Holography. *arXiv preprint arXiv:2505.06582* (2025).

- 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.
- Daniel Duckworth, Peter Hedman, Christian Reiser, Peter Zhizhin, Jean-François Thibert, Mario Lukić, Richard Szeliski, and Jonathan T Barron. 2024. Smerf: Streamable memory efficient radiance fields for real-time large-scene exploration. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–13.
- Daniel Glasner, Todd Zickler, and Anat Levin. 2014. A reflectance display. *ACM Trans. Graph.* 33, 4, Article 61 (July 2014), 12 pages. <https://doi.org/10.1145/2601097.2601140>
- Joseph W Goodman. 2005. *Introduction to Fourier optics*. Roberts and Company publishers.
- Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Yffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. 2019. The relightables: volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.* 38, 6, Article 217 (Nov. 2019), 19 pages. <https://doi.org/10.1145/3355089.3356571>
- Peter Hedman, Suhil Alisan, Richard Szeliski, and Johannes Kopf. 2017. Casual 3D photography. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–15.
- Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*. 1–11.
- Changwon Jang, Kiseung Bang, Minseok Chae, Byoungsoo Lee, and Douglas Lanman. 2024. Waveguide holography for 3D augmented reality glasses. *Nature Communications* 15, 1 (2024), 66.
- Koray Kavakli, Yuta Itoh, Hakan Urey, and Kaan Akşit. 2023a. Realistic defocus blur for multiplane computer-generated holography. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 418–426.
- Koray Kavakli, Liang Shi, Hakan Urey, Wojciech Matusik, and Kaan Akşit. 2023b. Multi-color Holograms improve Brightness in Holographic Displays. In *ACM SIGGRAPH ASIA 2023 Conference Proceedings*. ACM, Sydney, NSW, Australia, -. <https://doi.org/10.1145/3610548.3618135>
- Koray Kavakli, Hakan Urey, and Kaan Akşit. 2022. Learned holographic light transport. *Applied Optics* 61, 5 (2022), B50–B55.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.
- Dongyeon Kim, Seung-Woo Nam, Suyeon Choi, Jong-Mo Seo, Gordon Wetzstein, and Yoonchan Jeong. 2024. Holographic parallax improves 3D perceptual realism. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–13.
- Youngeun Kim, Wonjoon Jin, Sunghyun Cho, and Seung-Hwan Baek. 2023. Neural spectro-polarimetric fields. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.
- Zhao-Song Li, Chao Liu, Xin-Wei Li, Huan Yu, Di Wang, and Qiong-Hua Wang. 2025. Real-time holographic camera for obtaining real 3D scene hologram. *Light: Science & Applications* 14, 1 (2025), 74. <https://doi.org/10.1038/s41377-024-01730-9>
- Kexuan Liu, Jiachen Wu, Zehao He, and Liangcai Cao. 2023. 4K-DMDNet: diffraction model-driven network for 4K computer-generated holography. *Opto-Electronic Advances* 6, 5 (2023), 220135–1.
- Ilya Loshchilov and Frank Hutter. 2017. 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. <https://openreview.net/forum?id=Skq89Scxx>
- Andrew Maimone, Andreas Georgiou, and Joel S Kollin. 2017. Holographic near-eye displays for virtual and augmented reality. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–16.
- Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. 2024. Taming 3DGS: High-Quality Radiance Fields with Limited Resources. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. <https://doi.org/10.1145/3680528.3687694>
- Kyoji Matsushima and Sumio Nakahara. 2009. Extremely high-definition full-parallax computer-generated hologram created by the polygon-based method. *Applied optics* 48, 34 (2009), H54–H63.
- Kyoji Matsushima and Tomoyoshi Shimobaba. 2009. Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Optics express* 17, 22 (2009), 19662–19673.
- Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* (2019). <https://bmild.github.io/llff/>
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.
- Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. 2024. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. *arXiv preprint arXiv:2403.13806* (2024).
- Yifan Peng, Suyeon Choi, Nitish Padmanaban, and Gordon Wetzstein. 2020. Neural holography with camera-in-the-loop training. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.
- Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. 2021. DeRF: Decomposed Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14153–14161.
- Gernot Riegler and Vladlen Koltun. 2020. Free view synthesis. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*. Springer, 623–640.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. 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 18*. Springer, 234–241.
- Samuel Rota Bulò, Lorenzo Porzi, and Peter Kotschieder. 2024. Revising densification in gaussian splatting. In *European Conference on Computer Vision*. Springer, 347–362.
- Liang Shi, Beichen Li, Changil Kim, Petr Kellnhofer, and Wojciech Matusik. 2021. Towards real-time photorealistic 3D holography with deep neural networks. *Nature* 591, 7849 (2021), 234–239.
- Liang Shi, Beichen Li, and Wojciech Matusik. 2022. End-to-end learning of 3d phase-only holograms for holographic display. *Light: Science & Applications* 11, 1 (2022), 247.
- Tomoyoshi Shimobaba, Nobuyuki Masuda, and Tomoyoshi Ito. 2009. Simple and fast calculation algorithm for computer-generated hologram with wavefront recording plane. *Opt. Lett.* 34, 20 (Oct 2009), 3133–3135. <https://doi.org/10.1364/OL.34.003133>
- Shlomi Steinberg, Ravi Ramamoorthi, Benedikt Bitterli, Eugene d'Eon, Ling-Qi Yan, and Matt Pharr. 2024. A Generalized Ray Formulation For Wave-Optical Light Transport. *ACM Trans. Graph.* 43, 6, Article 241 (Nov. 2024), 15 pages. <https://doi.org/10.1145/3687902>
- Shlomi Steinberg, Pradeep Sen, and Ling-Qi Yan. 2022. Towards practical physical-optics rendering. *ACM Trans. Graph.* 41, 4, Article 132 (July 2022), 24 pages. <https://doi.org/10.1145/3528223.3530119>
- Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. 2025. FreeTimeGS: Free Gaussian Primitives at Anytime Anywhere for Dynamic Scene Reconstruction. In *CVPR*. <https://github.com/freetimegs>
- Beichen Wen, Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 2025. 3D Scene Generation: A Survey. *arXiv preprint arXiv:2505.05474* (2025).
- Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. 2024. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
- Han-Ju Yeom, Sanghoon Cheon, Kyunghye Choi, and Joongki Park. 2022. Efficient mesh-based realistic computer-generated hologram synthesis with polygon resolution adjustment. *ETRI Journal* 44, 1 (2022), 85–93.
- Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. 2019. Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets. *arXiv e-prints*, Article arXiv:1903.05662 (March 2019), arXiv:1903.05662 pages. <https://doi.org/10.48550/arXiv.1903.05662> arXiv:1903.05662 [cs.LG]
- Hyeonseung Yu, Youngrok Kim, Daeho Yang, Wontaek Seo, Yunhee Kim, Jong-Young Hong, Hoon Song, Geeyoung Sung, Younghun Sung, Sung-Wook Min, and Hong-Seok Lee. 2023. Deep learning-based incoherent holographic camera enabling acquisition of real-world holograms for holographic streaming system. *Nature Communications* 14, 1 (2023), 3534. <https://doi.org/10.1038/s41467-023-39329-0>
- Yicheng Zhan, Xiangjun Gao, Long Quan, and Kaan Akşit. 2025. Complex-Valued 2D Gaussian Representation for Computer-Generated Holography. *arXiv e-prints*, Article arXiv:2511.15022 (Nov. 2025), arXiv:2511.15022 pages. <https://doi.org/10.48550/arXiv.2511.15022> arXiv:2511.15022 [cs.CV]
- Yicheng Zhan, Qi Sun, Liang Shi, Wojciech Matusik, and Kaan Akşit. 2024. Configurable Holography: Towards Display and Scene Adaptation. *arXiv preprint arXiv:2405.01558* (2024).
- Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018a. 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, 586–595. <https://doi.org/10.1109/CVPR.2018.00068>
- Wenhui Zhang, Hao Zhang, and Guofan Jin. 2020. Band-extended angular spectrum method for accurate diffraction calculation in a wide propagation range. *Optics letters* 45, 6 (2020), 1543–1546.
- Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. 2024. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*. Springer, 327–345.

- Yuanzhe Zhang, Dewen Cheng, Yesheng Wang, Yongdong Wang, Yuefan Shan, Tong Yang, and Yongtian Wang. 2025. Real-time multi-depth holographic display using complex-valued neural network. *Optics Express* 33, 4 (2025), 7380–7395.
- Ya-Ping Zhang, Fan Wang, Ting-Chung Poon, Shuang Fan, and Wei Xu. 2018b. Fast generation of full analytical polygon-based computer-generated holograms. *Optics express* 26, 15 (2018), 19206–19224.
- Jinyu Zhao, Yusuke Monno, and Masatoshi Okutomi. 2022. Polarimetric multi-view inverse rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 7 (2022), 8798–8812.
- Chuanjun Zheng, Yicheng Zhan, Liang Shi, Ozan Cakmakci, and Kaan Akşit. 2024. Focal Surface Holographic Light Transport using Learned Spatially Adaptive Convolutions. In *SIGGRAPH Asia 2024 Technical Communications (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 28, 4 pages. <https://doi.org/10.1145/3681758.3697989>
- Chongli Zhong, Xinzhu Sang, Binbin Yan, Hui Li, Duo Chen, Xiujuan Qin, Shuo Chen, and Xiaoqian Ye. 2023. Real-time high-quality computer-generated hologram using complex-valued convolutional neural network. *IEEE Transactions on Visualization and Computer Graphics* (2023).

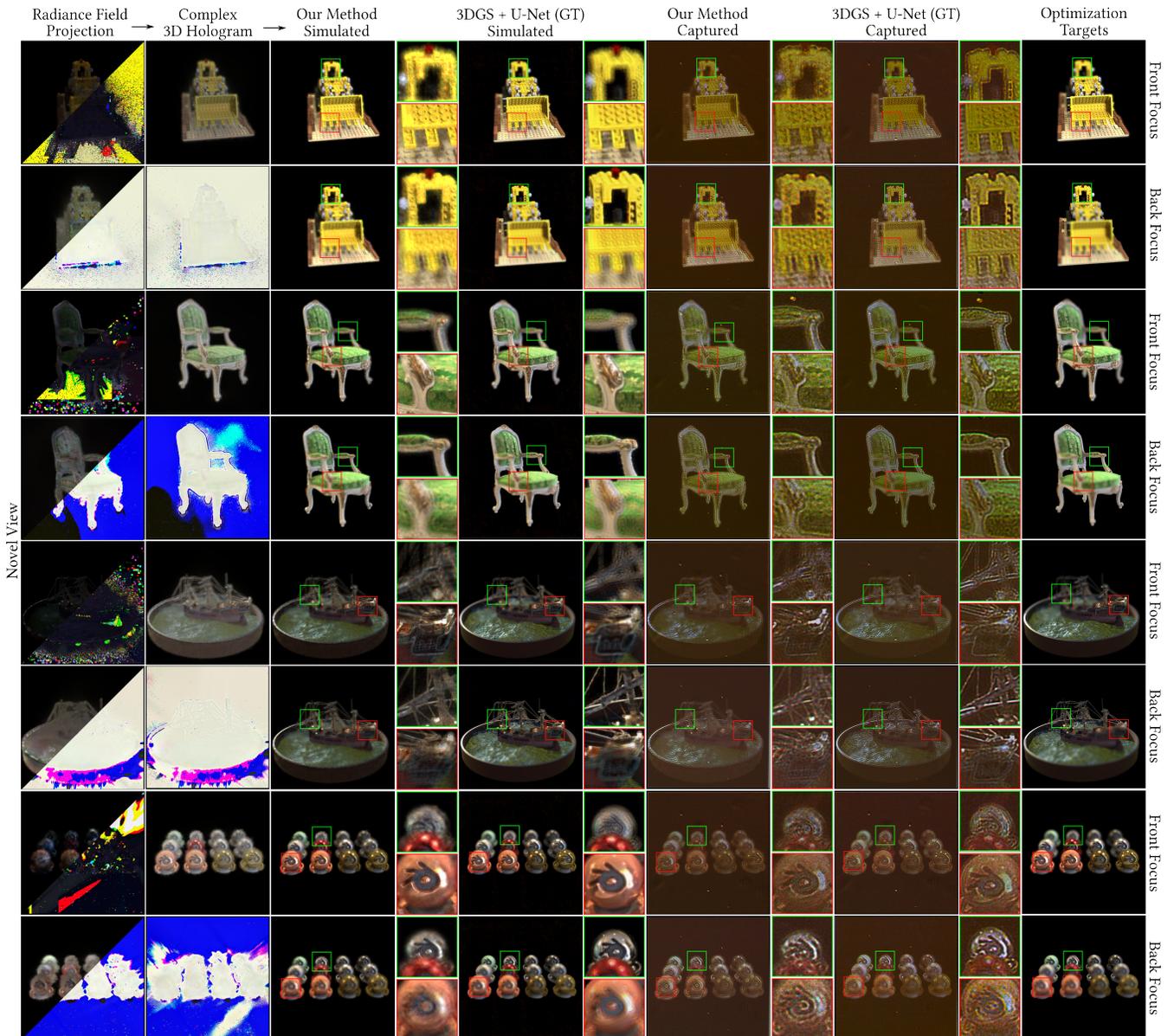


Fig. 8. Novel-view comparison between our method and the 3DGS + U-Net baseline on multiple scenes from the NeRF Synthetic dataset (lego, chair, ship, materials). Here GT represents the *Image Quality Ground Truth*. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present a side-by-side evaluation of our method’s simulated results and experimentally captured results against the baseline. The rightmost column displays the target images used as optimization objectives in our method.



Fig. 9. Novel-view comparison between our method and the 3DGS + U-Net baseline on multiple scenes from the LLFF dataset (flower, fern). Here GT represents the *Image Quality Ground Truth*. The central columns present a side-by-side evaluation of our method’s simulated results and experimentally captured results against the baseline. The rightmost column displays the target images used as optimization objectives in our method.



Fig. 10. Novel-view comparison of our method on multiple scene from the Mip-NeRF 360 Synthetic dataset (garden, kitchen). The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method and 3DGS + U-Net’s simulated results. The rightmost column displays the target images used as optimization objectives in our method.

# Supplementary: Complex-Valued Holographic Radiance Fields

YICHENG ZHAN, University College London, United Kingdom

DONG-HA SHIN, Pohang University of Science and Technology (POSTECH), South Korea

SEUNG-HWAN BAEK, Pohang University of Science and Technology (POSTECH), South Korea

KAAN AKŞIT, University College London, United Kingdom

## ACM Reference Format:

Yicheng Zhan, Dong-Ha Shin, Seung-Hwan Baek, and Kaan Akşit. 2024. Supplementary: Complex-Valued Holographic Radiance Fields. 1, 1 (March 2024), 19 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## A Hardware Image

Fig. 1 shows the photograph of the holographic display prototype 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 Spatial Light Modulator (SLM), the 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.

## B Wave Propagation

In this section, we provide additional insights into the implementation of our proposed complex-valued holographic radiance field.

### B.1 Wave Propagation From a Single 3D Gaussian Primitive

Each Gaussian  $\mathcal{G}_n$  can be considered as a small volume in  $\Omega$  that emits a coherent wavefield. Intuitively, we may approximate the total field from  $\mathcal{G}_n$  at hologram pixel  $P_i$ 's coordinate  $(x, y)$  by an integral of infinitesimal point contributions at the surface of the Gaussian. **To simplify the problem, if we only consider  $\mathcal{G}_n$ 's**

Authors' addresses: Yicheng Zhan, UCABY83@ucl.ac.uk, University College London, Gower Street, London, United Kingdom, WC1E 6BT; Dong-Ha Shin, 0218sdh@gmail.com, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro, Nam-gu, Pohang-si, South Korea, 37673; Seung-Hwan Baek, shwbaek@postech.ac.kr, Pohang University of Science and Technology (POSTECH), 77 Cheongam-ro, Nam-gu, Pohang-si, South Korea, 37673; Kaan Akşit, University College London, Gower Street, London, United Kingdom, k.aksit@ucl.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2024 Association for Computing Machinery.

XXXX-XXXX/2024/3-ART \$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

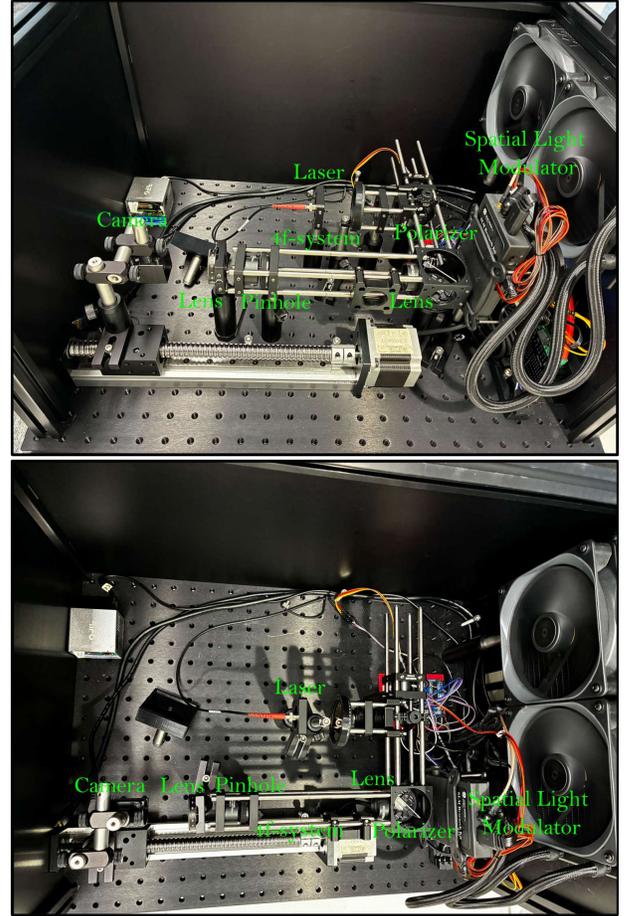


Fig. 1. The photograph of the holographic display prototype (Jasper JD7714) used in evaluating holograms generated by our model.

center at  $\mathbf{x}_n = (X_n, Y_n, Z_n)$ , the distance  $D_n$  from the  $\mathcal{G}_n$  to the hologram pixel at  $(x, y)$  is

$$D_n = \sqrt{(x - X_n)^2 + (y - Y_n)^2 + Z_n^2} \quad (1)$$

For a point light source emitting spherical waves, the phase at the hologram pixel is

$$\varphi_n^{\text{prop}} = \frac{2\pi}{\lambda} D_n \quad \text{mod } 2\pi \quad (2)$$

Under paraxial approximation, phase can be described using a quadratic function

$$\varphi_n^{\text{approx}} = \frac{2\pi}{\lambda} \left( Z_n + \frac{(x - X_n)^2 + (y - Y_n)^2}{2Z_n} \right) \quad \text{mod } 2\pi \quad (3)$$

The resulting complex field from Gaussian  $\mathcal{G}_n$  at the hologram pixel is

$$U_n = A_n \exp(j\varphi_n^{\text{prop}}), \quad A_n = c_n \alpha_n \quad (4)$$

where  $A_n$  includes the Gaussian's  $\alpha_n$  weighted amplitude, and  $c_n$  represents the inherent wave amplitude of the Gaussian. The full complex field integration of  $\mathcal{G}_n$  can be derived by expanding this point response to the Gaussian's spatial extent  $S_n$  and rotation  $\mathbf{R}_n$  by densely discretizing the surface of the Gaussian. This yields a volume of wavefront footprint recorded at the hologram plane  $P$ . Without considering the occlusion, the total field at the hologram pixel can be defined as:

$$U^{\text{total}} = \sum_{n=1}^N U_n \quad (5)$$

One may average over the sub-samples to compute the field at hologram pixel  $P_i$ , or keeping them explicit for higher accuracy modeling.

## B.2 Occlusion Aware Field Recording.

In a pure intensity world, to account for fore-ground occlusion, the frontmost Gaussian along the line of sight tends to occlude anything behind it, and people apply alpha-blending to represent transparent pixels. For simulating physical wave propagation in a hologram, occlusion is subtler—each "surface" can block or partially block wavefronts behind it. One approach to accurately model occlusion is to reconstruct a triangle surface mesh by connecting the neighboring points of Gaussians and performing ray casting from each point to the hologram plane  $P$ . The wavefronts carried by rays that hit the surface mesh are excluded from the hologram calculation. A binary visibility mask can be generated from this process to mimic foreground-occlusion at the hologram plane. However, both point-based wave simulation and ray-triangle detection of the entire scene are computationally expensive, making them a less favorable choice in real-time inference. As illustrated in Fig. 2, in our implementation, we use alpha-blending with transmittance to efficiently approximate occlusion effects, each 3D Gaussian  $\mathcal{G}_n$  can be considered as a small volume in  $\Omega$  that emits a coherent wavefield, rather than the more computationally expensive ray-casting approach described above.

## C Training Camera Viewpoint Sampling

During training, camera viewpoints are sampled differently depending on the dataset characteristics to ensure the learned scene representation supports consistent rendering across viewpoints within each dataset's coverage.

For the NeRF Synthetic dataset, we randomly sample camera poses facing towards the object within the Blender coordinate system, maintaining the same radial distance distribution as the original dataset while randomizing the azimuth and elevation angles. This generates a continuous distribution of training viewpoints around the object.

For the LLFF dataset, we directly use the existing camera viewpoints provided in the dataset, as the captured views already provide sufficient coverage of the indoor scenes with manageable motion parallax.

Complex-Valued Holographic Radiance Field

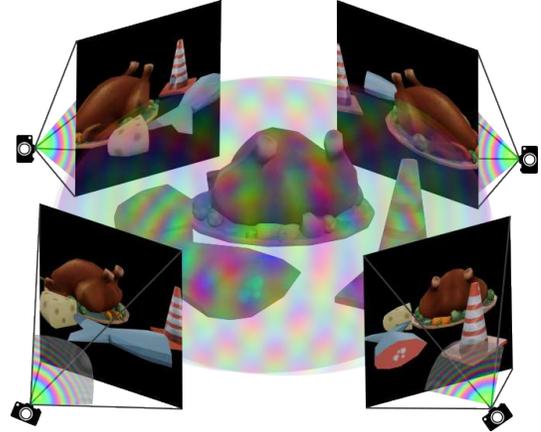


Fig. 2. Our method assumes the radiance field is a wave emitter, instead of a ray tracer.

For the Mip-NeRF 360 dataset, we employ a two-stage approach due to complex field discontinuities challenges arising from extreme motion parallax in the original wide-baseline captures. First, we train an intensity-based 3DGS scene using the provided dataset images. We uniformly sample 480 camera views covering a full 360° rotation, corresponding to 0.75° angular spacing for training, and evaluate novel-view rendering on the intermediate unseen viewpoints between sampled angles. We then use this trained 3DGS scene as a reference to randomly sample camera viewpoints facing towards the scene center, similar to the NeRF Synthetic sampling strategy. As a way round method, this approach generates training views with reduced inter-view motion parallax in the near scene compared to the original Mip-NeRF 360 captures, mitigating the severe complex field discontinuities that would otherwise cause substantial image quality degradation, but it does not help with the far field.

## D Differentiable Complex-Valued Gaussian Rendering

In this section, we provide the pseudo code and a detailed exposition of both the forward computation and the backward gradient derivation for our Differentiable Complex-Valued Rasterizer.

### D.1 Notation

- $\mathbf{x}_n \in \mathbb{R}^3$  - 3D Gaussian center positions in world space
- $\mathbf{x}_n^{\text{proj}} \in \mathbb{R}^2$  - Projected 2D Gaussian centers in screen space
- $\mu_x, \mu_y$  - Components of the 2D mean position
- $\Sigma \in \mathbb{R}^{3 \times 3}$  - 3D covariance matrix in world space
- $\Sigma' \in \mathbb{R}^{2 \times 2}$  - 2D covariance matrix in screen space
- $\Sigma'^{-1}$  - Elements of the inverse 2D covariance matrix (where  $i, j \in \{0, 1\}$ )
- $d_x = x - \mu_x, d_y = y - \mu_y$  - Distance from pixel to Gaussian center
- $\rho_{n,l}$  - Plane assignment probability of Gaussian  $n$  for plane  $l$  (controls contribution to each rendering plane)
- $\varphi_n$  - Phase value for Gaussian  $n$  (controls complex field phase)

- $\alpha_n$  - Opacity value for Gaussian  $n$
- $\alpha_n^{\text{base}}$  - Base opacity value before applying the Gaussian function
- $T_n$  - Accumulated transmittance at Gaussian  $n$  (product of previous transparency values)
- $z_{\text{norm}}$  - Normalized depth value between 0 and 1

## D.2 Optimization and Densification Algorithm

Our optimization and densification algorithms extend the standard 3D Gaussian Splatting (3DGS) [Kerbl et al. 2023] approach to handle complex-valued parameters and multi-plane assignments. The key modifications include phase parameter optimization, plane probability learning with Straight-Through Estimator (STE), and complex-valued loss computation as summarized in Algorithm 1.

---

### Algorithm 1 Optimization and Densification for Complex-Valued 3DGS

---

**Require:**  $w, h$ : width and height of the training images  
**Require:**  $L$ : number of depth planes for multi-plane rendering

```

1:  $M \leftarrow \text{SfM Points}$  ▷ Positions
2:  $S, C, \alpha \leftarrow \text{InitAttributes}()$  ▷ Scales, Amplitudes, Opacities
3:  $\varphi \leftarrow \text{InitPhase}()$  ▷ Phase Parameters
4:  $\rho' \leftarrow \text{InitPlaneProbs}(L)$  ▷ Plane Assignment Logits
5:  $i \leftarrow 0$  ▷ Iteration Count
6: while not converged do
7:    $V, I \leftarrow \text{SampleTrainingView}()$  ▷ Camera  $V$  and Image
8:    $\rho \leftarrow \text{STE}(\rho')$  ▷ Hard Assignment via STE
9:    $U_{\text{complex}} \leftarrow \text{ComplexRasterize}(M, S, C, \alpha, \varphi, \rho, V)$  ▷ Alg. 2
10:   $P \leftarrow \text{PropagateToHologram}(U_{\text{complex}})$  ▷ Multi-plane ASM
11:   $I_{\text{recon}} \leftarrow \text{BackPropagate}(P)$  ▷ Reconstruct Images
12:   $\mathcal{L} \leftarrow \text{ComplexLoss}(I, I_{\text{recon}})$  ▷  $\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{SSIM}}$ 
13:   $M, S, C, \alpha, \varphi, \rho' \leftarrow \text{Adam}(\nabla \mathcal{L})$  ▷ Backprop & Step
14:  if IsRecentIteration() then
15:    for all Gaussians  $(\mu, \Sigma, c, \alpha, \varphi, \rho')$  in  $(M, S, C, \alpha, \varphi, \rho')$  do
16:      if  $\alpha < \epsilon$  or IsTooLarge( $\mu, \Sigma, c, \alpha$ ) then ▷ Pruning
17:        RemoveGaussian()
18:      end if
19:      if  $\nabla_{\mu} \mathcal{L} > \tau_p$  then ▷ Densification
20:        if  $\|S\| > \tau_s$  then ▷ Over-reconstruction
21:          SplitGaussian( $\mu, \Sigma, c, \alpha, \varphi, \rho'$ )
22:        else ▷ Under-reconstruction
23:          CloneGaussian( $\mu, \Sigma, c, \alpha, \varphi, \rho'$ )
24:        end if
25:      end if
26:    end for
27:  end if
28:   $i \leftarrow i + 1$ 
29: end while

```

---

The key extensions to the standard algorithm include: (1) Phase parameter  $\varphi$  initialization with random values in  $[0, 2\pi)$ ; (2) Plane assignment probabilities  $\rho'$  initialized uniformly and converted to hard assignments via STE during forward pass; (3) Complex-valued rasterization producing multi-plane complex fields; (4) Hologram generation through multi-plane Angular Spectrum Method (ASM) propagation; (5) Phase-aware densification where both split and clone operations preserve phase relationships by adding small random perturbations to maintain coherent interference patterns.

## D.3 Details of the Rasterizer

Our tile-based rasterizer extends the standard 3DGS approach to handle complex-valued rendering across multiple depth planes. The algorithm processes Gaussians in back-to-front order within each

tile to ensure correct alpha blending for wave-optical effects, as detailed in Algorithm 2.

---

### Algorithm 2 Complex-Valued Tile-Based Rasterization

---

**Require:**  $w, h$ : width and height of the image to rasterize  
**Require:**  $M, S$ : Gaussian means and covariances in world space  
**Require:**  $C, \alpha, \varphi, \rho$ : Amplitudes, opacities, phases, and plane assignments  
**Require:**  $V$ : view configuration of current camera  
**Require:**  $L$ : number of depth planes

```

1: function COMPLEXRASTERIZE( $w, h, M, S, C, \alpha, \varphi, \rho, V$ )
2:   CullGaussians( $M, V$ ) ▷ Frustum Culling
3:    $M', S' \leftarrow \text{ScreenspaceGaussians}(M, S, V)$  ▷ Transform
4:    $T \leftarrow \text{CreateTiles}(w, h)$  ▷  $16 \times 16$  Tiles
5:    $I, \mathcal{K} \leftarrow \text{DuplicateWithDepthKeys}(M', T)$  ▷ Back-to-Front Keys
6:    $\mathcal{K}_s, \mathcal{I}_s \leftarrow \text{SortByKey}(I, \mathcal{K})$  ▷ Depth Sort
7:    $\mathcal{R} \leftarrow \text{IdentifyTileRanges}(T, \mathcal{K}_s)$  ▷ Tile Ranges
8:    $U_{\text{real}}, U_{\text{imag}} \leftarrow \text{InitComplexCanvas}(L, w, h)$  ▷ Multi-plane Output
9:   for all Tiles  $t$  in  $T$  do
10:     for all Planes  $l$  in  $L$  do
11:       for all Pixels  $\text{pix}$  in  $t$  do
12:          $T_{\text{acc}} \leftarrow 1.0$  ▷ Transmittance
13:          $\text{real}_{\text{acc}}, \text{imag}_{\text{acc}} \leftarrow 0.0, 0.0$  ▷ Complex Accumulation
14:          $\text{range} \leftarrow \text{GetTileRange}(\mathcal{R}, t)$ 
15:         for  $g$  in  $\text{range}$  do ▷ Back-to-Front Order
16:            $w_{\text{plane}} \leftarrow \rho_{g,l}$  ▷ Plane Assignment
17:           if  $w_{\text{plane}} < \epsilon_{\text{plane}}$  then
18:             continue
19:           end if
20:            $\alpha_{\text{eff}} \leftarrow \alpha_g \cdot \exp(-\text{power}_{g,\text{pix}}) \cdot w_{\text{plane}}$  ▷ Effective Alpha
21:            $\text{scale} \leftarrow C_g \cdot \alpha_{\text{eff}} \cdot T_{\text{acc}}$  ▷ Complex Scale
22:            $\text{real}_{\text{acc}} \leftarrow \text{real}_{\text{acc}} + \text{scale} \cdot \cos(\varphi_g)$  ▷ Real Component
23:            $\text{imag}_{\text{acc}} \leftarrow \text{imag}_{\text{acc}} + \text{scale} \cdot \sin(\varphi_g)$  ▷ Imaginary Component
24:            $T_{\text{acc}} \leftarrow T_{\text{acc}} \cdot (1 - \alpha_{\text{eff}})$  ▷ Update Transmittance
25:           if  $T_{\text{acc}} < \epsilon_T$  then
26:             break ▷ Early Termination
27:           end if
28:         end for
29:          $U_{\text{real}}[l, \text{pix}] \leftarrow \text{real}_{\text{acc}}$ 
30:          $U_{\text{imag}}[l, \text{pix}] \leftarrow \text{imag}_{\text{acc}}$ 
31:       end for
32:     end for
33:   end for
34:   return  $U_{\text{real}} + j \cdot U_{\text{imag}}$ 
35: end function

```

---

The rasterizer incorporates several key innovations: (1) Back-to-front depth sorting using inverted depth keys to ensure proper alpha blending for wave propagation; (2) Multi-plane rendering where each Gaussian contributes to planes based on learned assignment probabilities  $\rho_{g,l}$ ; (3) Complex field accumulation using amplitude-weighted trigonometric functions  $C_g \cdot \cos(\varphi_g)$  and  $C_g \cdot \sin(\varphi_g)$  for real and imaginary components; (4) Plane-specific early termination when transmittance falls below threshold  $\epsilon_T$ ; (5) Efficient shared memory usage for tile-based processing with complex-valued intermediate storage.

## D.4 Forward Pass

**D.4.1 Complex Field Generation** For each pixel  $(x, y)$  in each plane  $\Pi_l$ , the complex field  $U$  is:

$$U_{\Pi_l}(x, y) = \text{real}_{\Pi_l}(x, y) + i \cdot \text{imag}_{\Pi_l}(x, y) \quad (6)$$

Where for each Gaussian  $\mathcal{G}_n$ :

$$\begin{aligned} \text{real}_{\Pi_l}(x, y) &= \sum_n \mathbf{c}_n \cdot \boldsymbol{\alpha}_n \cdot T_n \cdot \cos(\varphi_n) \\ \text{imag}_{\Pi_l}(x, y) &= \sum_n \mathbf{c}_n \cdot \boldsymbol{\alpha}_n \cdot T_n \cdot \sin(\varphi_n) \end{aligned} \quad (7)$$

#### D.4.2 Transmittance Calculation

$$T_n = \prod_{j=1}^{n-1} (1 - \alpha_j) \quad (8)$$

#### D.4.3 Alpha Calculation

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_n^{\text{base}} \cdot \mathcal{G}_n^{\text{proj}} \cdot \boldsymbol{\rho}_{n,l} \quad (9)$$

#### D.4.4 Gaussian Projection (Exponent Term)

$$\begin{aligned} \mathcal{G}_n^{\text{proj}} &= \exp(-0.5 \cdot (d_x \cdot (\Sigma'_{00}{}^{-1} \cdot d_x + \Sigma'_{10}{}^{-1} \cdot d_y) \\ &\quad + d_y \cdot (\Sigma'_{01}{}^{-1} \cdot d_x + \Sigma'_{11}{}^{-1} \cdot d_y))) \end{aligned} \quad (10)$$

Where  $d_x = x - \mu_x$  and  $d_y = y - \mu_y$ .

### D.5 Backward Pass

**D.5.1 Gradient Flow Overview** For each parameter  $\theta$ , the gradient follows the chain rule from the final loss  $\mathcal{L}$  to the parameter:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \theta} \quad (11)$$

where  $\mathcal{L}$  is the loss and  $P$  is the hologram output.

#### D.5.2 Detailed Gradient Derivation for Each Parameter

##### (1) Gradient for rotation parameters (related to $\mathbf{R}_n$ )

The gradient with respect to rotation parameters follows the chain rule:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{R}_n} &= \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \boldsymbol{\alpha}_n} \cdot \frac{\partial \boldsymbol{\alpha}_n}{\partial \mathcal{G}_n^{\text{proj}}} \cdot \frac{\partial \mathcal{G}_n^{\text{proj}}}{\partial \text{term}} \\ &\quad \cdot \frac{\partial \text{term}}{\partial \Sigma'^{-1}} \cdot \frac{\partial \Sigma'^{-1}}{\partial \Sigma'} \cdot \frac{\partial \Sigma'}{\partial \Sigma} \cdot \frac{\partial \Sigma}{\partial \mathbf{R}_n} \cdot \frac{\partial \Sigma}{\partial \mathbf{q}} \end{aligned} \quad (12)$$

Key components include:

- $\frac{\partial \mathcal{L}}{\partial P}$ : Gradient of loss with respect to hologram output, provided by optimization algorithm.
- $\frac{\partial P}{\partial U_{\Pi_l}}$ : Derivative of hologram with respect to complex field, computed via band-limited propagation.
- $\frac{\partial U_{\Pi_l}}{\partial \boldsymbol{\alpha}_i} = \mathbf{c}_i \cdot T_i \cdot e^{j \cdot \varphi_i} - \sum_{k=i+1}^N \mathbf{c}_k \cdot \boldsymbol{\alpha}_k \cdot T_k \cdot \frac{\partial T_k}{\partial \boldsymbol{\alpha}_i} \cdot e^{j \cdot \varphi_k}$  where  $\frac{\partial T_k}{\partial \boldsymbol{\alpha}_i} = -T_k / (1 - \alpha_i)$  for  $i < k$ .
- $\frac{\partial \boldsymbol{\alpha}_n}{\partial \mathcal{G}_n^{\text{proj}}} = \boldsymbol{\alpha}_n^{\text{base}} \cdot \boldsymbol{\rho}_{n,l}$ , from  $\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_n^{\text{base}} \cdot \mathcal{G}_n^{\text{proj}} \cdot \boldsymbol{\rho}_{n,l}$ .
- $\frac{\partial \mathcal{G}_n^{\text{proj}}}{\partial \text{term}} = -0.5 \cdot \mathcal{G}_n^{\text{proj}}$ , from  $\mathcal{G}_n^{\text{proj}} = \exp(-0.5 \cdot \text{term})$ .
- $\frac{\partial \text{term}}{\partial \Sigma'^{-1}} = \Delta \cdot \Delta^T$  where  $\Delta = (p - \mathbf{x}_n^{\text{proj}})$  is the pixel-mean difference.
- $\frac{\partial \Sigma'^{-1}}{\partial \Sigma'} = -\Sigma'^{-1} \otimes \Sigma'^{-1}$ , following matrix inverse derivative.
- $\frac{\partial \Sigma'_{ij}}{\partial \Sigma_{kl}} = \sum_{m,n} J_{im} \mathbf{R}_{mk} \mathbf{R}_{nl} J_{jn}$ , where  $J$  is the projection Jacobian and  $\mathbf{R}$  the rotation matrix.
- $\frac{\partial \Sigma}{\partial \mathbf{R}_{ij}} = \frac{\partial (\mathbf{R} \mathbf{S} \mathbf{S}^T \mathbf{R}^T)}{\partial \mathbf{R}_{ij}} = E_{ij} \cdot \mathbf{S} \cdot \mathbf{S}^T \cdot \mathbf{R}^T + \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{S}^T \cdot E_{ji}^T$ , where  $E_{ij}$  is a standard basis matrix.

(j)  $\frac{\partial \mathbf{R}_n}{\partial \mathbf{q}} = \frac{\partial \mathbf{R}}{\partial \mathbf{q}}$ , where  $\mathbf{q}$  is the quaternion representation of rotation.

##### (2) Gradient for $\mathbf{x}_n$ (Gaussian centers)

The gradient for  $\mathbf{x}_n$  passes through multiple paths in the computation graph:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_n} &= \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \boldsymbol{\alpha}_n} \cdot \frac{\partial \boldsymbol{\alpha}_n}{\partial \mathcal{G}_n^{\text{proj}}} \cdot \frac{\partial \mathcal{G}_n^{\text{proj}}}{\partial \text{term}} \\ &\quad \cdot \frac{\partial \text{term}}{\partial \text{diff}} \cdot \frac{\partial \text{diff}}{\partial \mathbf{x}_n^{\text{proj}}} \cdot \frac{\partial \mathbf{x}_n^{\text{proj}}}{\partial \mathbf{x}_n^{\text{cam}}} \cdot \frac{\partial \mathbf{x}_n^{\text{cam}}}{\partial \mathbf{x}_n} \end{aligned} \quad (13)$$

Key components include:

- $\frac{\partial U_{\Pi_l}}{\partial \boldsymbol{\alpha}_n}$ : As described above.
- $\frac{\partial \boldsymbol{\alpha}_n}{\partial \mathcal{G}_n^{\text{proj}}} = \boldsymbol{\alpha}_n^{\text{base}} \cdot \boldsymbol{\rho}_{n,l}$
- $\frac{\partial \mathcal{G}_n^{\text{proj}}}{\partial \text{term}} = -0.5 \cdot \mathcal{G}_n^{\text{proj}}$
- $\frac{\partial \text{term}}{\partial \text{diff}} = \Sigma'^{-1} \cdot \text{diff} + \text{diff} \cdot \Sigma'^{-1}$ , where  $\text{diff} = (p - \mathbf{x}_n^{\text{proj}})$  is the pixel-mean difference
- $\frac{\partial \text{diff}}{\partial \mathbf{x}_n^{\text{proj}}} = -I$  (negative identity matrix)
- $\frac{\partial \mathbf{x}_n^{\text{proj}}}{\partial \mathbf{x}_n^{\text{cam}}}$ : Projection Jacobian
- $\frac{\partial \mathbf{x}_n^{\text{cam}}}{\partial \mathbf{x}_n} = \mathbf{R}$  (camera rotation matrix)

Critically, for pixels  $(x, y)$  and Gaussian centers  $(\mu_x, \mu_y)$ , we set  $d_x = x - \mu_x$  and  $d_y = y - \mu_y$ . The gradient components are:

$$\begin{aligned} \frac{\partial \text{term}}{\partial \mu_x} &= -(-d_x \cdot \text{inv}_{00} - d_y \cdot \text{inv}_{10}) \\ \frac{\partial \text{term}}{\partial \mu_y} &= -(-d_x \cdot \text{inv}_{01} - d_y \cdot \text{inv}_{11}) \end{aligned} \quad (14)$$

Note the negative sign in the implementation that results from  $\frac{\partial (x - \mu_x)}{\partial \mu_x} = -1$ . This is correctly captured in our final gradient computation.

##### (3) Gradient for covariance matrix elements

The covariance gradient calculation is performed using the chain rule and the matrix inverse derivative formula:

$$\frac{\partial \mathcal{L}}{\partial \Sigma} = -\Sigma^{-1} \cdot \frac{\partial \mathcal{L}}{\partial \Sigma^{-1}} \cdot \Sigma^{-1} \quad (15)$$

For the 2D covariance matrix elements, we compute:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Sigma_{00}} &= -(bt_{00} \cdot \text{tmp}_{00} + bt_{01} \cdot \text{tmp}_{10}) \\ \frac{\partial \mathcal{L}}{\partial \Sigma_{01}} &= -(bt_{00} \cdot \text{tmp}_{01} + bt_{01} \cdot \text{tmp}_{11}) \\ \frac{\partial \mathcal{L}}{\partial \Sigma_{10}} &= -(bt_{10} \cdot \text{tmp}_{00} + bt_{11} \cdot \text{tmp}_{10}) \\ \frac{\partial \mathcal{L}}{\partial \Sigma_{11}} &= -(bt_{10} \cdot \text{tmp}_{01} + bt_{11} \cdot \text{tmp}_{11}) \end{aligned} \quad (16)$$

where:

- $bt$  is the transpose of the inverse covariance matrix
- $\text{tmp}$  represents intermediate gradient values from the quadratic form

This ensures that the negative sign from the matrix inverse derivative is properly incorporated.

##### (4) Gradient for $\mathbf{S}_n$ (Gaussian scales)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}_n} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \alpha_n} \cdot \frac{\partial \alpha_n}{\partial \mathcal{G}_n^{\text{proj}}} \cdot \frac{\partial \mathcal{G}_n^{\text{proj}}}{\partial \text{term}} \cdot \frac{\partial \text{term}}{\partial \Sigma'^{-1}} \cdot \frac{\partial \Sigma'^{-1}}{\partial \Sigma'} \cdot \frac{\partial \Sigma'}{\partial \Sigma} \cdot \frac{\partial \Sigma}{\partial \mathbf{S}_n} \quad (17)$$

(a) Components from  $\frac{\partial \mathcal{L}}{\partial P}$  through  $\frac{\partial \Sigma'}{\partial \Sigma}$  as previously described.

(b)  $\frac{\partial \Sigma_{ij}}{\partial \mathbf{S}_{n,k}} = 2 \cdot \mathbf{S}_{n,k} \cdot \mathbf{R}_{i,k} \cdot \mathbf{R}_{j,k}$  for anisotropic case.

#### (5) Gradient for $\mathbf{c}_n$ (Gaussian amplitudes)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{c}_n} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \mathbf{c}_n} \quad (18)$$

(a)  $\frac{\partial U_{\Pi_l}}{\partial \mathbf{c}_n} = \alpha_n \cdot T_n \cdot e^{j \cdot \varphi_n}$ , directly contributing to the complex field.

In the implementation, this becomes:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{c}_n} = \alpha_n \cdot T_{\text{before}} \cdot (\cos(\varphi_n) \cdot \frac{\partial \mathcal{L}}{\partial \text{real}} + \sin(\varphi_n) \cdot \frac{\partial \mathcal{L}}{\partial \text{imag}}) \quad (19)$$

#### (6) Gradient for $\varphi_n$ (Gaussian phases)

$$\frac{\partial \mathcal{L}}{\partial \varphi_n} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \varphi_n} \quad (20)$$

(a)  $\frac{\partial U_{\Pi_l}}{\partial \varphi_n} = j \cdot \mathbf{c}_n \cdot \alpha_n \cdot T_n \cdot e^{j \cdot \varphi_n}$ , from the derivative of the complex exponential.

This expands to:

$$\frac{\partial \mathcal{L}}{\partial \varphi_n} = \alpha_n \cdot T_{\text{before}} \cdot \mathbf{c}_n \cdot (-\sin(\varphi_n) \cdot \frac{\partial \mathcal{L}}{\partial \text{real}} + \cos(\varphi_n) \cdot \frac{\partial \mathcal{L}}{\partial \text{imag}}) \quad (21)$$

Note the negative sign on the sine term, which comes from the derivative of  $\cos(\varphi_n)$ .

#### (7) Gradient for $\alpha_n^{\text{base}}$ (Gaussian base opacities)

$$\frac{\partial \mathcal{L}}{\partial \alpha_n^{\text{base}}} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \alpha_n} \cdot \frac{\partial \alpha_n}{\partial \alpha_n^{\text{base}}} \quad (22)$$

(a)  $\frac{\partial U_{\Pi_l}}{\partial \alpha_n}$ : As previously described.

(b)  $\frac{\partial \alpha_n}{\partial \alpha_n^{\text{base}}} = \mathcal{G}_n^{\text{proj}} \cdot \rho_{n,l}$  from  $\alpha_n = \alpha_n^{\text{base}} \cdot \mathcal{G}_n^{\text{proj}} \cdot \rho_{n,l}$ .

In the implementation, this is calculated as:

$$\frac{\partial \mathcal{L}}{\partial \alpha_n^{\text{base}}} = \exp(\text{power}) \cdot \rho_{n,l} \cdot \frac{\partial \mathcal{L}}{\partial \alpha_n} \quad (23)$$

#### (8) Gradient for $\rho_{n,l}$ (plane assignment probabilities)

$$\frac{\partial \mathcal{L}}{\partial \rho_{n,l}} = \frac{\partial \mathcal{L}}{\partial P} \cdot \frac{\partial P}{\partial U_{\Pi_l}} \cdot \frac{\partial U_{\Pi_l}}{\partial \alpha_n} \cdot \frac{\partial \alpha_n}{\partial \rho_{n,l}} \quad (24)$$

(a)  $\frac{\partial U_{\Pi_l}}{\partial \alpha_n}$ : As previously described.

(b)  $\frac{\partial \alpha_n}{\partial \rho_{n,l}} = \alpha_n^{\text{base}} \cdot \mathcal{G}_n^{\text{proj}}$

In the implementation, the gradient for plane probabilities is:

$$\frac{\partial \mathcal{L}}{\partial \rho_{n,l}} = \alpha_n^{\text{base}} \cdot \exp(\text{power}) \cdot \frac{\partial \mathcal{L}}{\partial \alpha_n} \quad (25)$$

**D.5.3 Multi-Plane Transmittance Calculation** For multiple planes, the transmittance is calculated separately for each plane  $l$ :

$$T_n^{(l)} = \prod_{j=1}^{n-1} (1 - \alpha_j^{(l)}), \quad \text{where } \alpha_j^{(l)} = \alpha_j^{\text{base}} \cdot \mathcal{G}_j^{\text{proj}} \cdot \rho_{j,l} \quad (26)$$

The gradient of transmittance for plane  $l$  is:

$$\frac{\partial T_k^{(l)}}{\partial \alpha_n^{(l)}} = \begin{cases} -T_k^{(l)} / (1 - \alpha_n^{(l)}) & \text{if } n < k \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

This approach ensures each plane's rendering is treated independently, with plane weights determining which planes a Gaussian contributes to. The CUDA implementation computes gradients per-pixel and per-Gaussian for each plane, with gradients accumulated atomically across all planes.

## E Ablation Study of CUDA operations

This section conduct ablation study on the contribution of our two key method components: tile-based complex-valued rasterization and FFT-based layer propagation. We measure performance using *torch.profiler*, reporting GPU time. Each iteration is wrapped by explicit profiling scopes, and the two components are isolated as timing windows; all CUDA kernels executed within each window are collected via hierarchical traversal and accumulated using self-device time to avoid double-counting. Per-kernel GPU times are averaged over 50 iterations to obtain a stable per-step runtime breakdown, without inserting any explicit CUDA synchronization that would perturb normal execution.

Fig. 3 presents a detailed profiling analysis of the computational breakdown for each rendering step. Averaged by 50 steps, the complex-valued splatting and forward recording step completes in 10.11 ms, with tile-based splatting (SplatTileCuda) accounting for 5.18 ms (51.2%), making it the dominant operation. Additional costs include batch matrix operations (1.71 ms, 16.9%) and FFT operations for band-limited propagation (1.44 ms, 14.3%).

The inverse propagation step completes in 9.00 ms, dominated by FFT-based complex-to-complex transformations (4.43 ms, 49.2%) for multi-plane propagation. Matrix multiplications contribute 2.27 ms (25.3%), while tensor roll operations for FFT alignment require 1.87 ms (20.7%). Together, tile-based rasterization and FFT-based propagation constitute 9.61 ms of the total 19.11 ms pipeline (50.3%), demonstrating that these two techniques are the primary computational contributors enabling real-time performance. The remaining overhead from tensor operations, sorting, and memory management represents 9.50 ms (49.7%).

## F Gaussian Distribution And Phase Characteristics Analysis

This section presents a quantitative analysis of the learned complex-valued holographic radiance field, focusing on phase behavior, spatial frequency content, and their relationship to Gaussian primitive properties.

### F.1 Phase and Amplitude Distributions

The learned hologram phase exhibits a structured distribution that lies between globally smooth and fully random patterns, with a

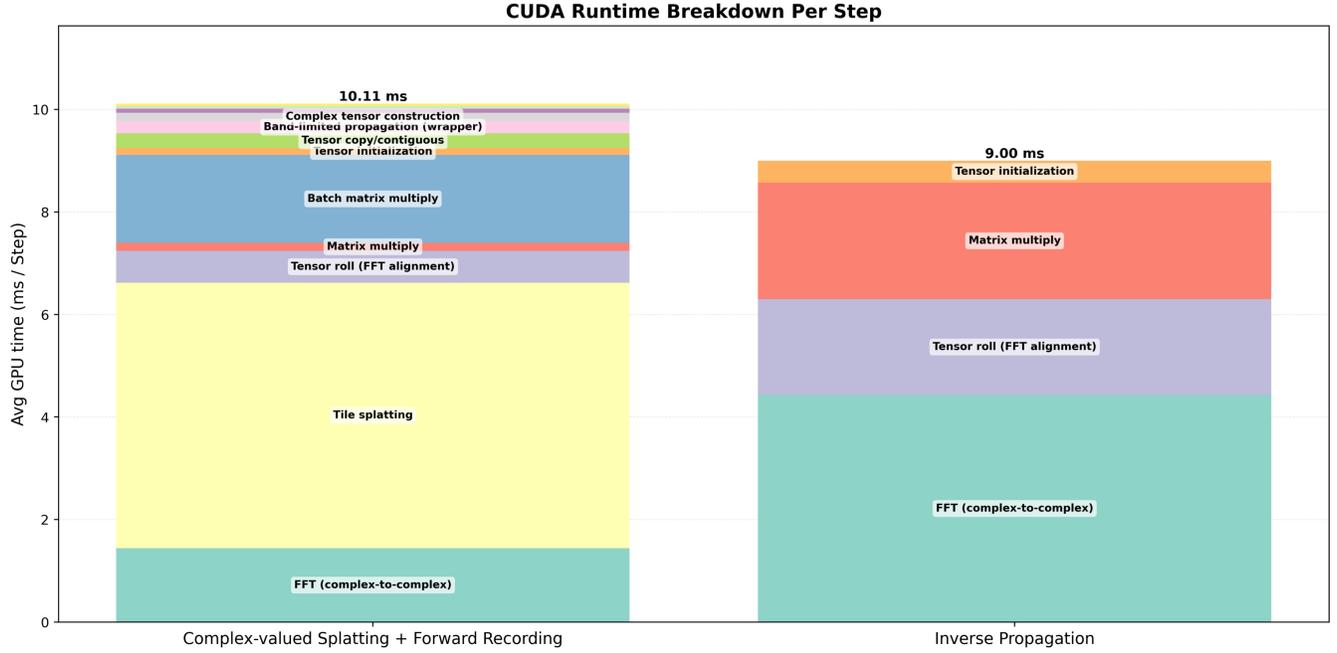


Fig. 3. Ablation study of CUDA runtime breakdown per step.

clear bias toward spatial smoothness. As shown in Fig. 4(j), the phase histogram is bimodal with peaks near 0 and  $2\pi$ , indicating non-uniform but organized phase statistics.

This behavior arises from the coherent superposition of multiple complex-valued Gaussian primitives. Each Gaussian acts as a spatially localized, low-pass basis function, enforcing continuous phase variation within its finite support. Consequently, individual primitives cannot generate high-frequency, pixel-wise phase fluctuations.

The final phase field is formed by summing many such locally smooth contributions, each associated with an independently learned phase parameter. Their interference introduces structured, higher-frequency variations while preserving local continuity, yielding phase patterns that are irregular yet not fully random.

Two factors jointly govern this behavior: (1) the Gaussian primitives impose local smoothness due to their finite spatial support; (2) phase parameters  $\varphi_n$  are learned per primitive rather than per pixel, inducing spatially correlated phase variations. As illustrated in Fig. 4(b), the resulting phase map exhibits extended coherent regions with sharper transitions primarily at overlap regions and object boundaries.

The amplitude distribution Fig. 4(k) is strongly right-skewed, with most values near zero and a long tail at higher amplitudes. This reflects the sparsity of the representation, where only a subset of Gaussians contribute significantly at each spatial location. Low amplitudes correspond to background regions, while the tail captures areas of constructive interference. The amplitude map Fig. 4(a) confirms that energy is concentrated in object regions.

## F.2 Spatial Coherence and Gradient Analysis

The low-pass nature of Gaussian primitives induces spatially smooth phase structure, as shown by the phase gradient magnitude in Fig. 4(e). Gradients remain low across most of the hologram plane, increasing primarily at object boundaries and depth discontinuities. This behavior follows directly from the continuous Gaussian projections.

Interference between overlapping Gaussians introduces localized high-frequency variations. The amplitude gradient map (Fig. 4(d)) shows sharp transitions concentrated at object edges, where multiple primitives with differing phases overlap.

## F.3 Spatial Frequency Characteristics

Fourier-domain analysis Fig. 4(c,f) reveals a spectrum dominated by low-frequency components, consistent with smooth spatial variations from Gaussian primitives. Nevertheless, substantial mid-to-high frequency energy is present near the center, encoding fine interference details required for accurate holographic reconstruction. The spatial frequency profile decays monotonically from DC.

## F.4 Gaussian Primitive Characteristics

The learned Gaussian scale distribution Fig. 4(g) is heavily concentrated in the range 0.00–0.05, with exponential decay toward larger scales. This reflects the densification strategy's preference for small primitives in geometrically complex regions, enabling high-resolution modeling while maintaining efficiency, this is consistent with traditional 3DGS. The dominance of small-scale Gaussians further promotes local phase coherence due to their limited spatial support.

The opacity distribution Fig. 4(h) is bimodal, with peaks near 0.0 and 1.0, indicating specialization toward either transparent or opaque states. The mean opacity near 0.5 reflects a balance between coverage and transmittance, enabling effective modeling of wave-front occlusion via accumulated transmittance.

### F.5 Depth Distribution and Plane Assignment

The depth distribution Fig. 4(i) shows Gaussians clustering around predefined depth planes, confirming effective learning of plane assignment probabilities. This validates the effectiveness of the STE-based plane assignment.

For multi-plane settings, the plane assignment statistics (Fig. 4(l)) show balanced utilization across planes (e.g., 51.4% vs. 48.6%), indicating no collapse to a single plane. As shown in the first row’s zoomed insets of Fig. 5, the high-frequency details of Gaussians for front and back focus are gathered at separate depth planes, demonstrating the effectiveness of the plane splitting of our method.

### F.6 Relationship to Holographic Reconstruction

Fig. 5 demonstrates how the learned parameters in our method will affect reconstruction. Multi-plane amplitude and phase maps exhibit coherent phase structure within objects and localized discontinuities at depth boundaries, consistent with the statistical analysis.

During forward recording, plane-specific complex fields are aggregated via band-limited ASM propagation ( $H_{Z_l}$ ). Inverse propagation ( $H_{-Z_l}$ ) produces reconstructed intensities with natural defocus blur. In-focus regions remain sharp, while out-of-focus areas exhibit smooth diffraction blur without structured artifacts existed in the existing learned CGH methods.

### F.7 Novel View Synthesis Progression

Fig. 6 demonstrates the behavior of learned Gaussian primitives under continuous camera trajectory variation, validating our implicit geometric learning mechanism.

The first two rows show Plane 1 and Plane 2 complex-field projections (amplitude and phase). Despite using fixed plane assignment probabilities  $\rho$  learned during training, the projected complex fields maintain correct geometric appearance across novel views. This occurs because:

- (1) **Occlusion handling through alpha blending:** Gaussians that become occluded at novel angles are naturally attenuated by the transmittance product  $\prod_{j=1}^{n-1} (1 - \alpha_j)$ .
- (2) **Multi-view learned assignments:** The plane probabilities  $\rho_n$  are optimized jointly over the full training viewpoint distribution, enabling them to function collectively rather than being viewpoint-specific.
- (3) **Geometric awareness:** While no Gaussian is explicitly supervised with a ground-truth 3D depth during forward recording, their collective behavior—through learned  $\rho$  and  $\alpha$  parameters—produces geometrically consistent results, which empirically generalize well to continuously interpolated novel views.

To further clarify this behavior, consider a single Gaussian primitive in our representation. Each Gaussian is parameterized by both

a plane assignment probability  $\rho$  and an opacity  $\alpha$  (Eq. 9). These parameters are learned through multi-view reconstruction supervision and are fixed after training, making the Gaussian intrinsically view-independent and directly reusable for arbitrary novel viewpoints. While prior methods do not introduce an explicit plane assignment parameter  $\rho$ , the role of  $\alpha$  in our formulation is consistent with recent work [Choi et al. 2025], where opacity similarly governs occlusion and contribution strength.

The final column (Novel View 120) demonstrates robustness under large rotation angles, where the camera viewpoint transitions from one side of the chair to the opposite side. At this extreme angle, many Gaussians that contributed significantly at earlier views become heavily occluded; nevertheless, the reconstruction remains correct because other Gaussians that were previously attenuated now contribute more strongly. The zoomed insets reveal that individual Gaussians maintain consistent intrinsic properties (amplitude and phase) across viewpoints, while their effective contributions vary naturally due to occlusion. This validates that our method achieves view-consistent holographic encoding through learned scene parameters rather than per-view recalculation. The forward-recorded hologram (Row 3) and reconstructed intensities (Row 4) further confirm that fixed plane assignments, combined with proper occlusion modeling, enable correct rendering throughout the camera trajectory.

## G Extra Simulated And Captured Result

This section presents additional experimental results demonstrating the effectiveness of our complex-valued 3D Gaussian splatting approach across diverse scenes. We provide comprehensive comparisons on both synthetic and real-world datasets, including scenes from the NeRF Synthetic dataset (lego in Fig. 7, chair in Fig. 8, ship in Fig. 9, materials in Fig. 10, and hotdog in Fig. 11) and the LLFF dataset (flower in Fig. 12, fern in Fig. 13, trex in Fig. 14, and room in Fig. 15). Each figure showcases the complete pipeline from radiance field projections and complex 3D hologram rendering to both simulated and experimentally captured results, demonstrating consistent quality across different scene types and lighting conditions.

## References

- Suyeon Choi, Brian Chao, Jacqueline Yang, Manu Gopakumar, and Gordon Wetzstein. 2025. Gaussian Wave Splatting for Computer-Generated Holography. *arXiv preprint arXiv:2505.06582* (2025).
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

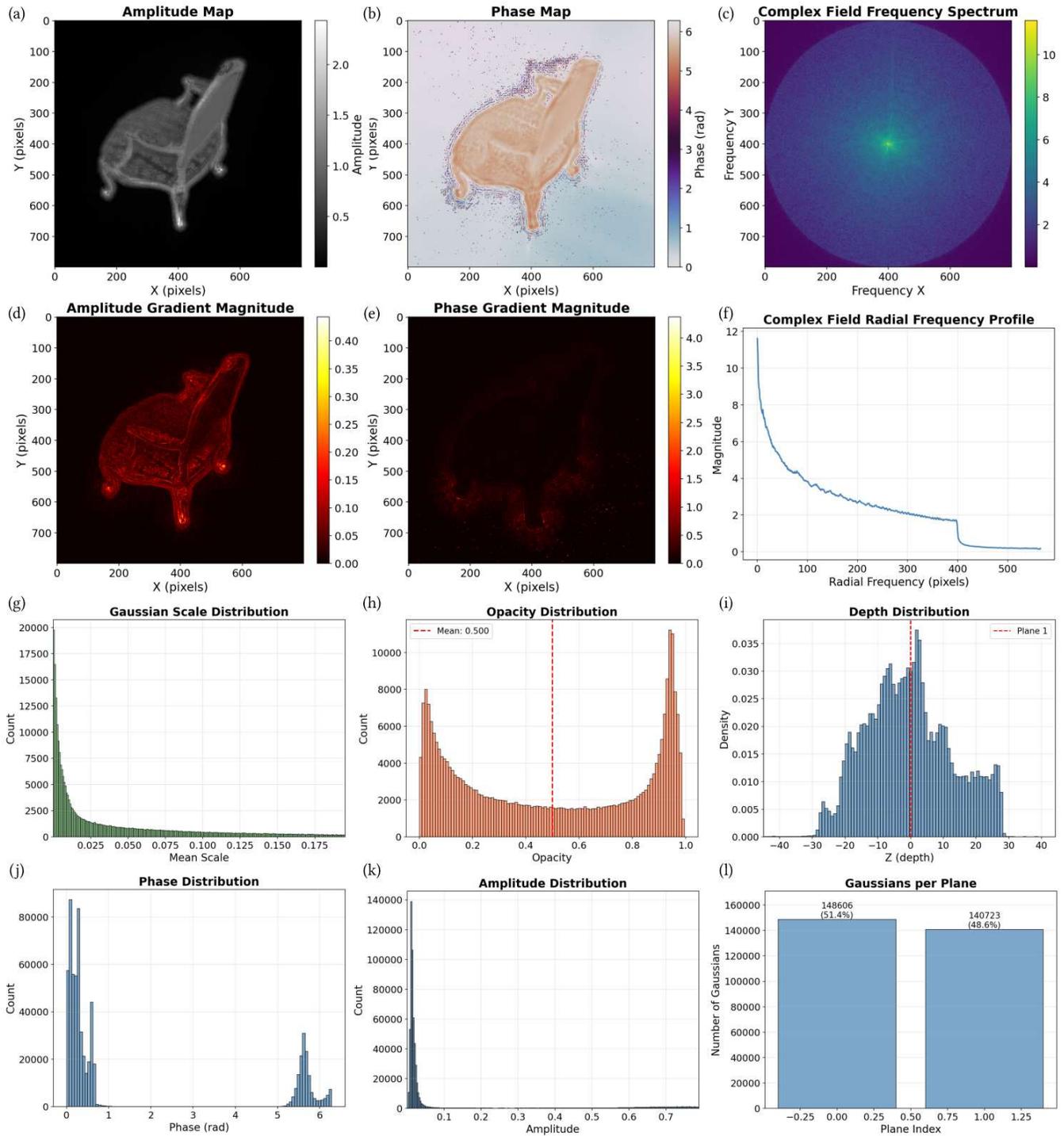


Fig. 4. Statistical analysis of learned complex-valued holographic radiance field properties. (a) Amplitude map. (b) Phase map. (c) Fourier-domain energy distribution. (d) Amplitude gradient magnitude. (e) Phase gradient magnitude. (f) Spatial frequency profile. (g) Gaussian scale distribution. (h) Opacity distribution. (i) Depth distribution. (j) Phase distribution. (k) Amplitude distribution. (l) Gaussian assignment across two depth planes.

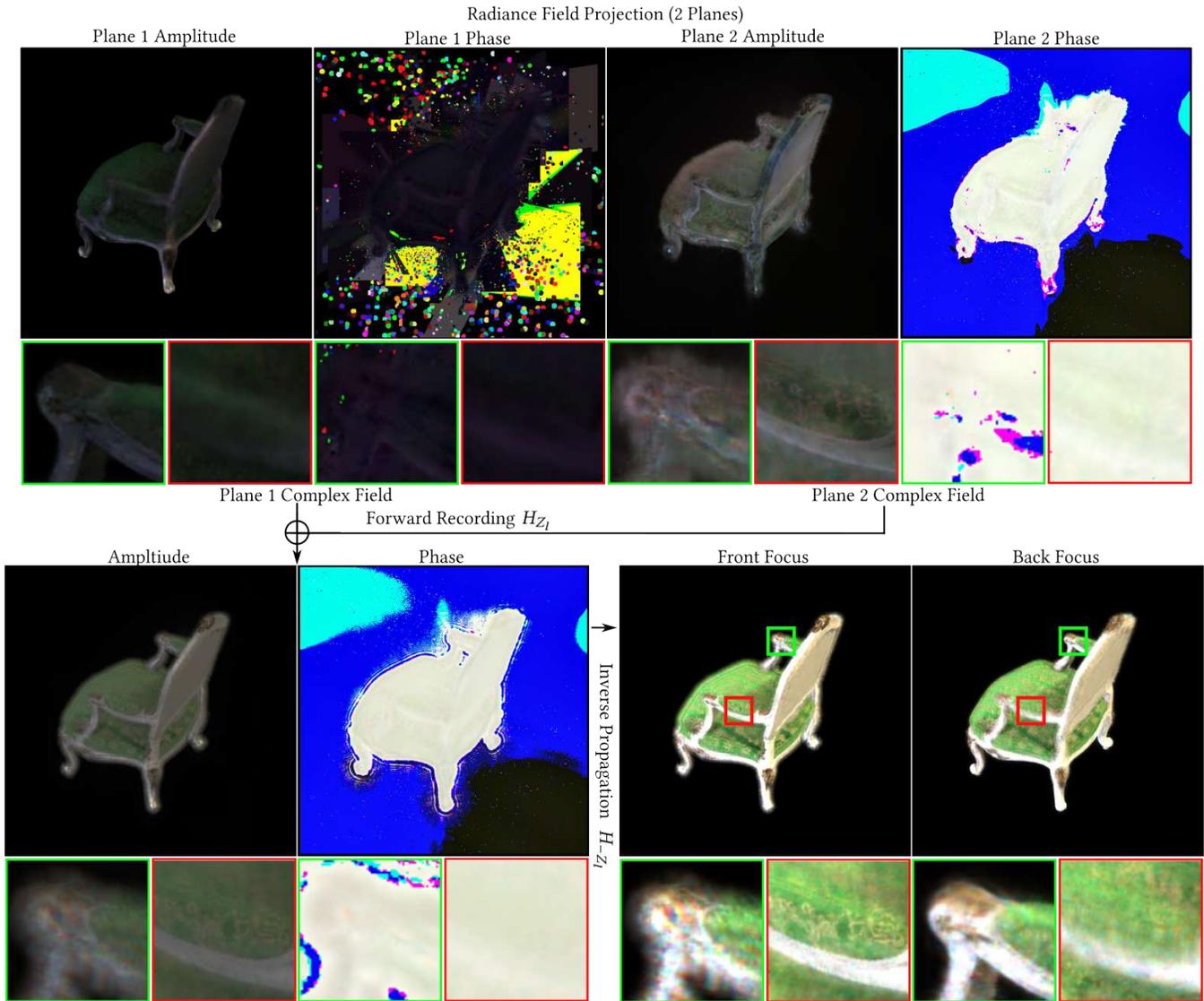


Fig. 5. Visualization of complex-valued holographic radiance field rendering pipeline and holographic reconstruction. **Top:** Multi-plane radiance field projections showing amplitude and phase maps for 2 Depth Planes. **Bottom Left** Forward recording process combines plane-specific complex fields via band-limited ASM ( $H_{Z_l}$ ) to generate the final hologram. **Bottom Right:** Reconstructed Intensities for front and back focus via Inverse Propagation ( $H_{-Z_l}$ ).

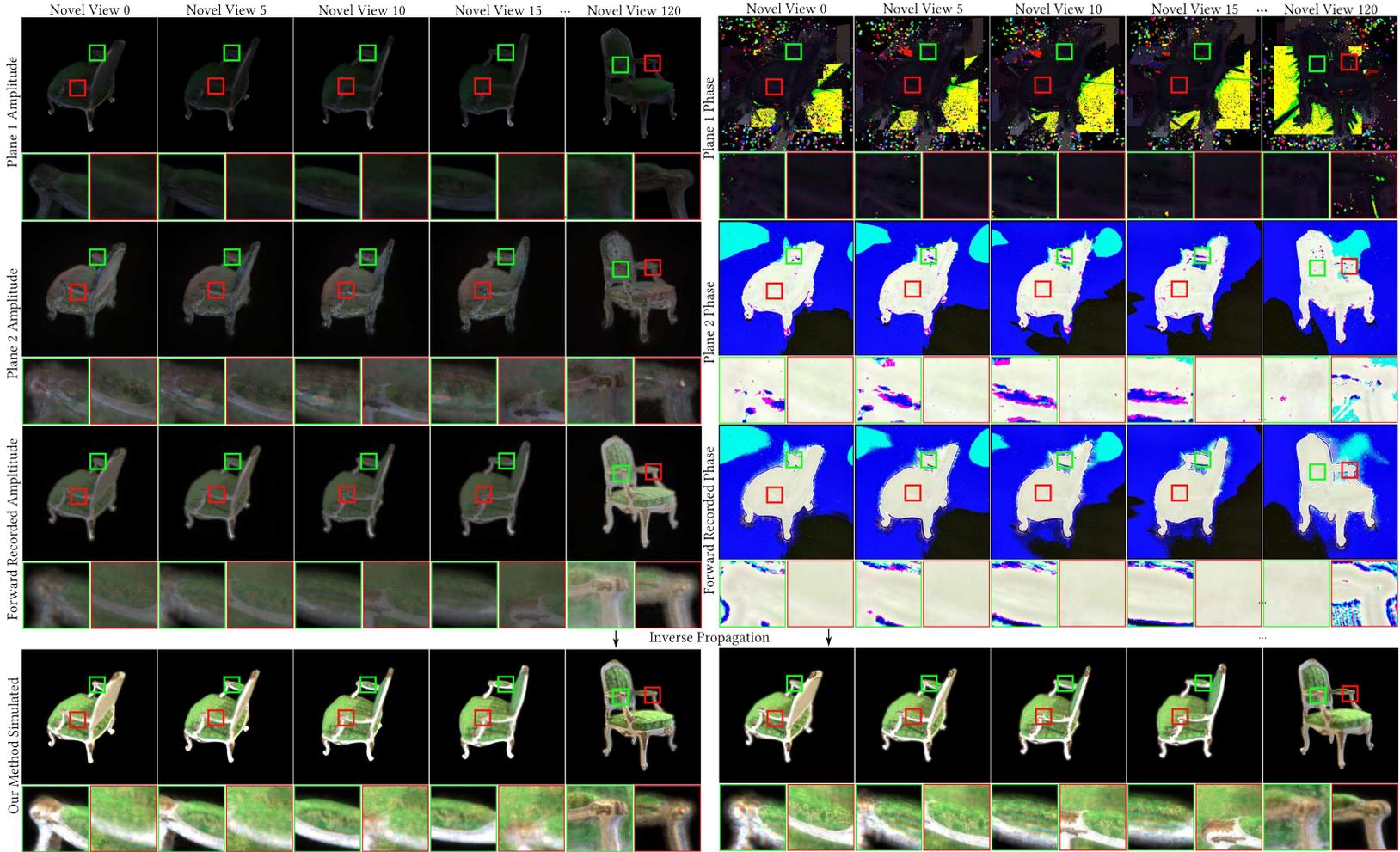


Fig. 6. Gaussian primitive contributions under continuous novel-view variation. Columns show novel viewpoints sampled along a continuous camera trajectory. **Row 1:** Plane 1 complex-field projections from Trained 3D complex-valued Gaussians, visualized as amplitude (left) and phase (right). **Row 2:** Plane 2 complex-field projections. **Row 3:** Forward-recorded hologram formed by coherently combining both depth-plane fields via band-limited ASM propagation. **Row 4:** Reconstructed intensities via inverse propagation, showing front- and back-focus responses.

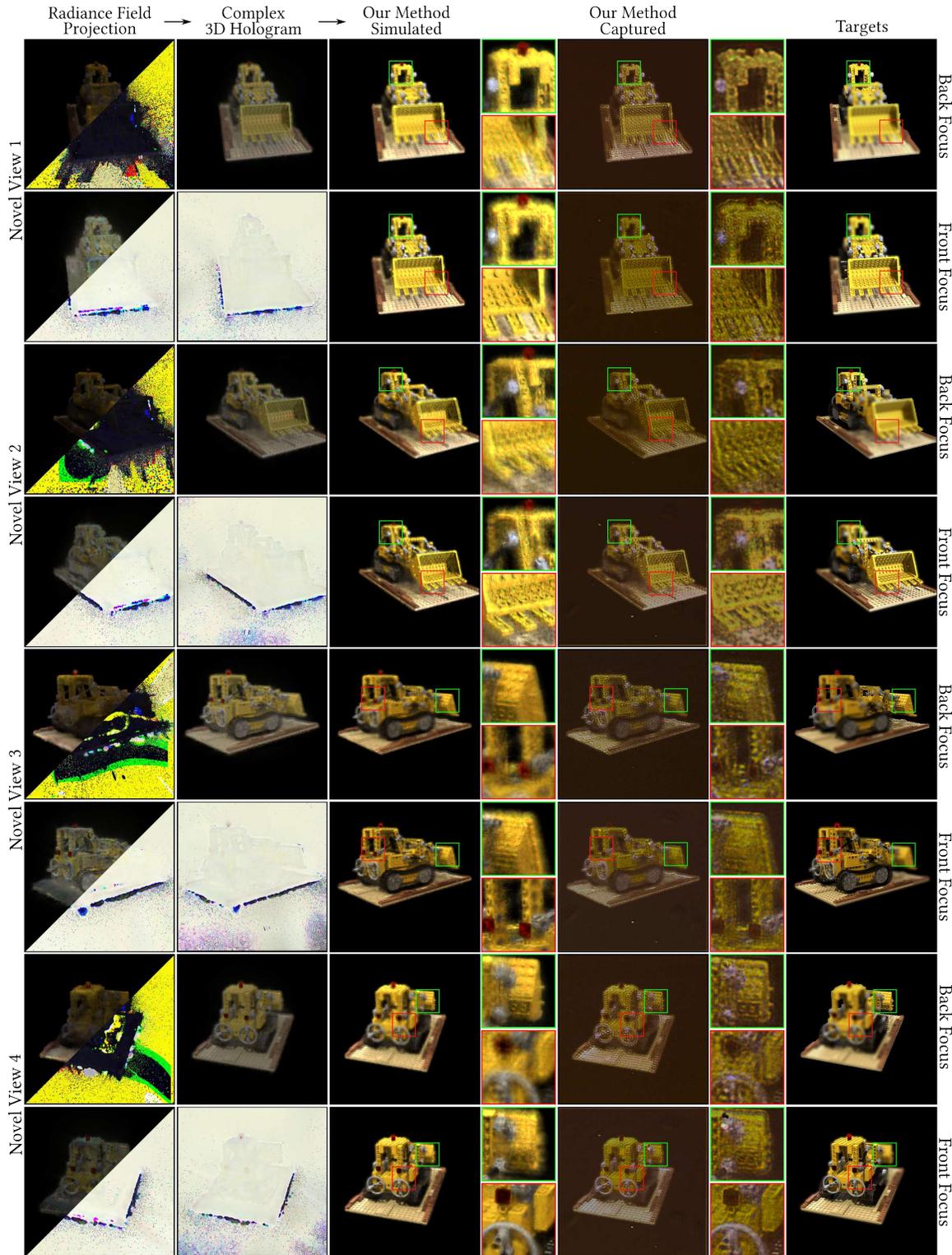


Fig. 7. Extra novel-view comparison of our method on lego scene from the NeRF Synthetic dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method’s simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

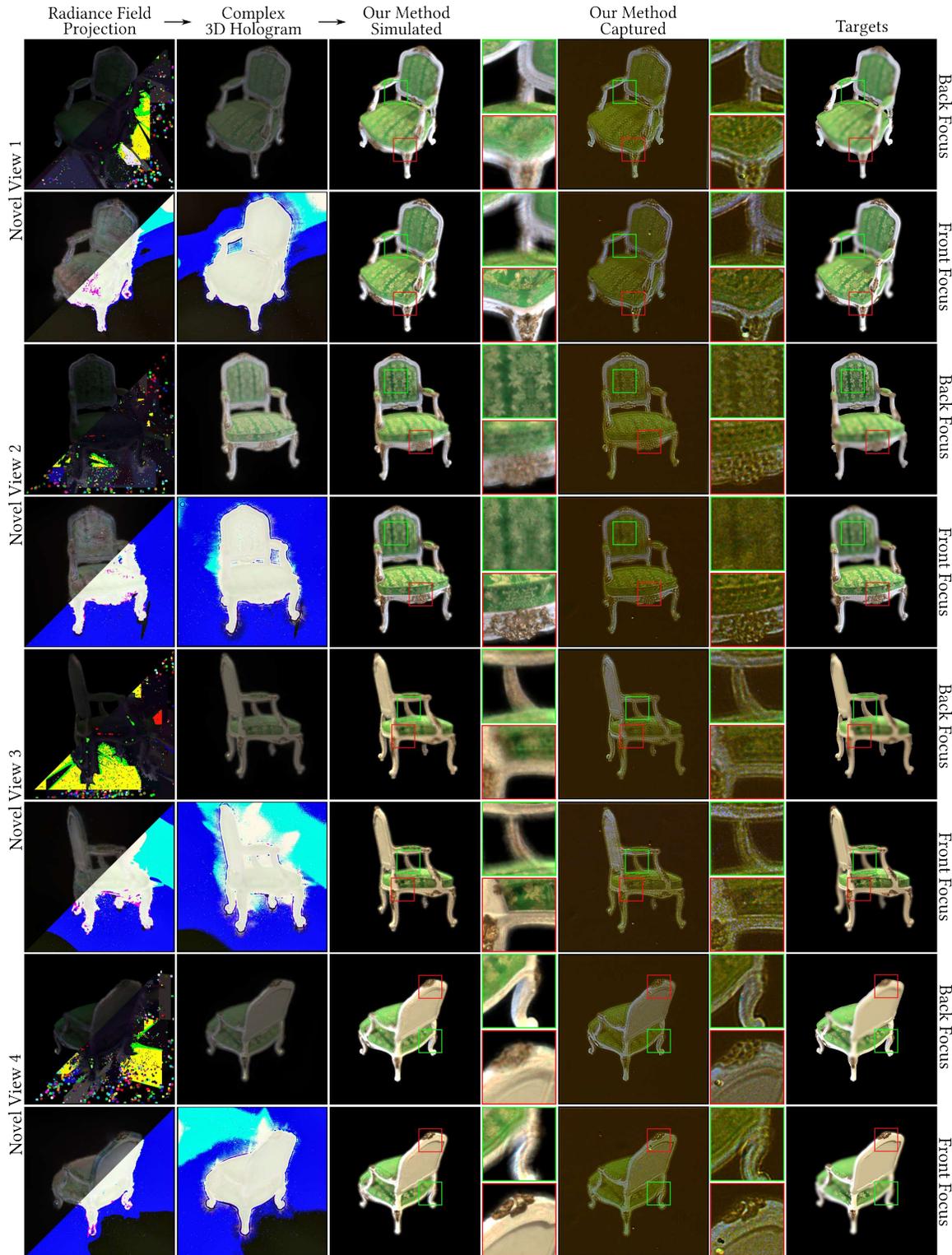


Fig. 8. Extra novel-view comparison of our method on chair scene from the NeRF Synthetic dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method’s simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

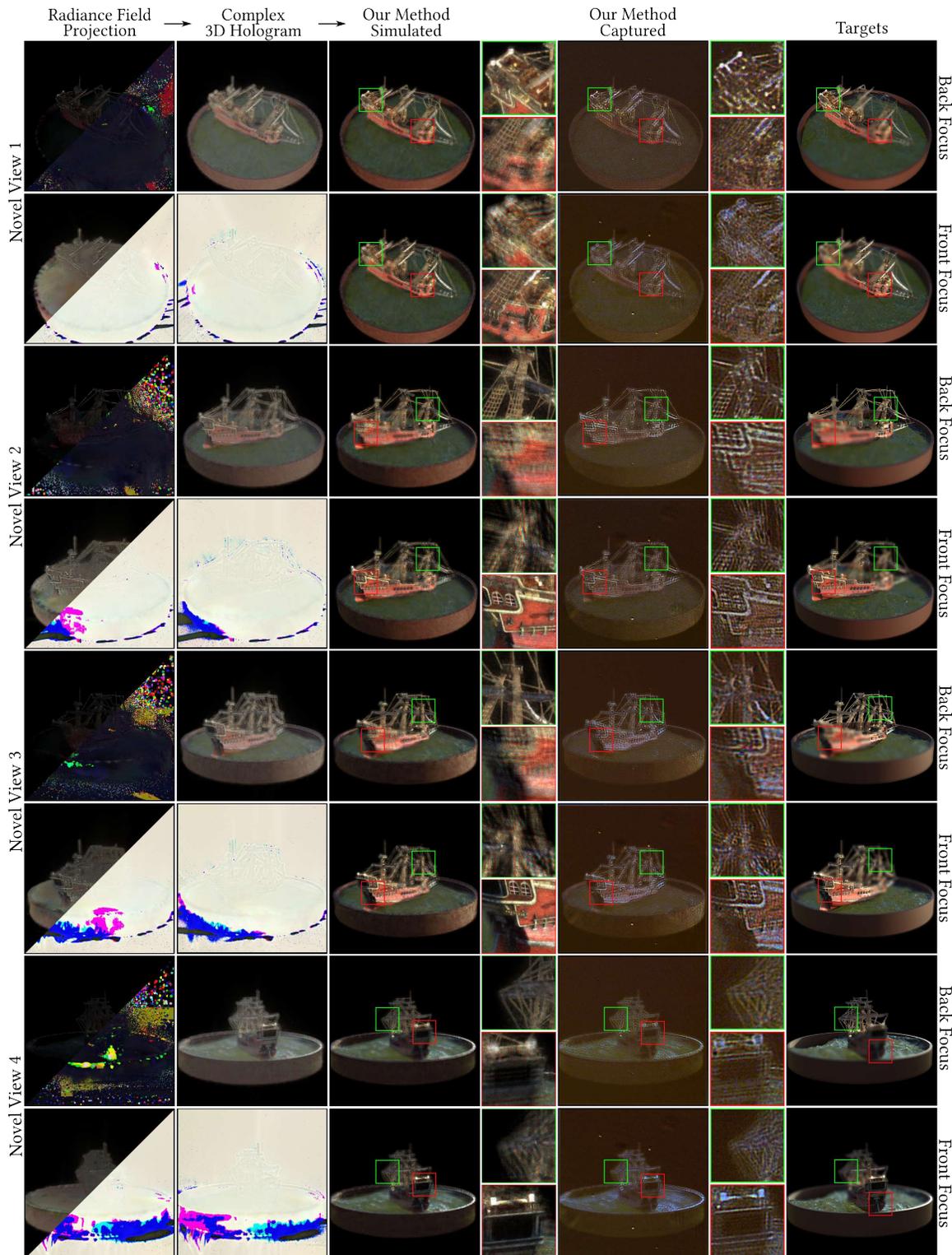


Fig. 9. Extra novel-view comparison of our method on ship scene from the NeRF Synthetic dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method's simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

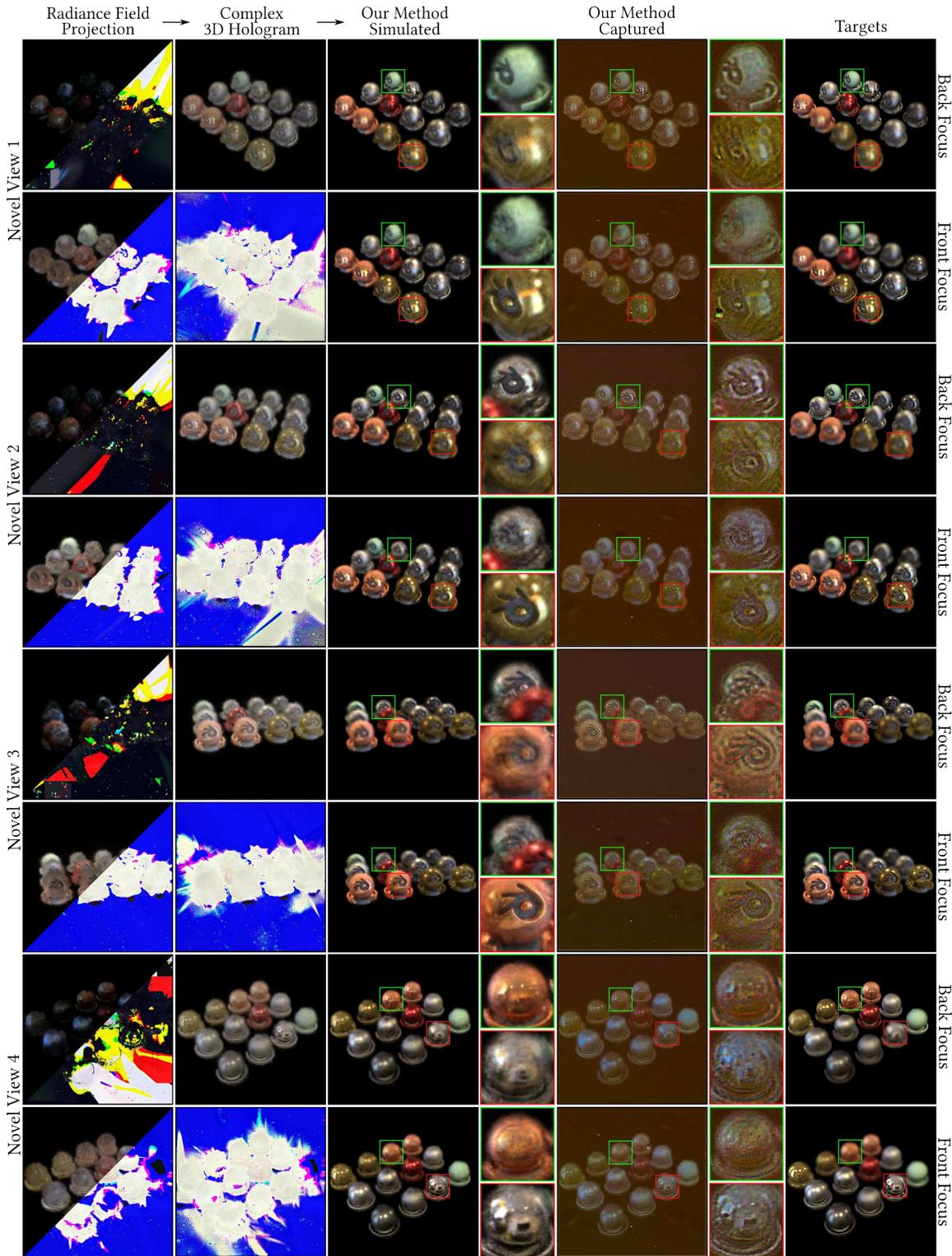


Fig. 10. Extra novel-view comparison of our method on materials scene from the NeRF Synthetic dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method’s simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.



Fig. 11. Extra novel-view comparison of our method on hotdog scene from the NeRF Synthetic dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method's simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

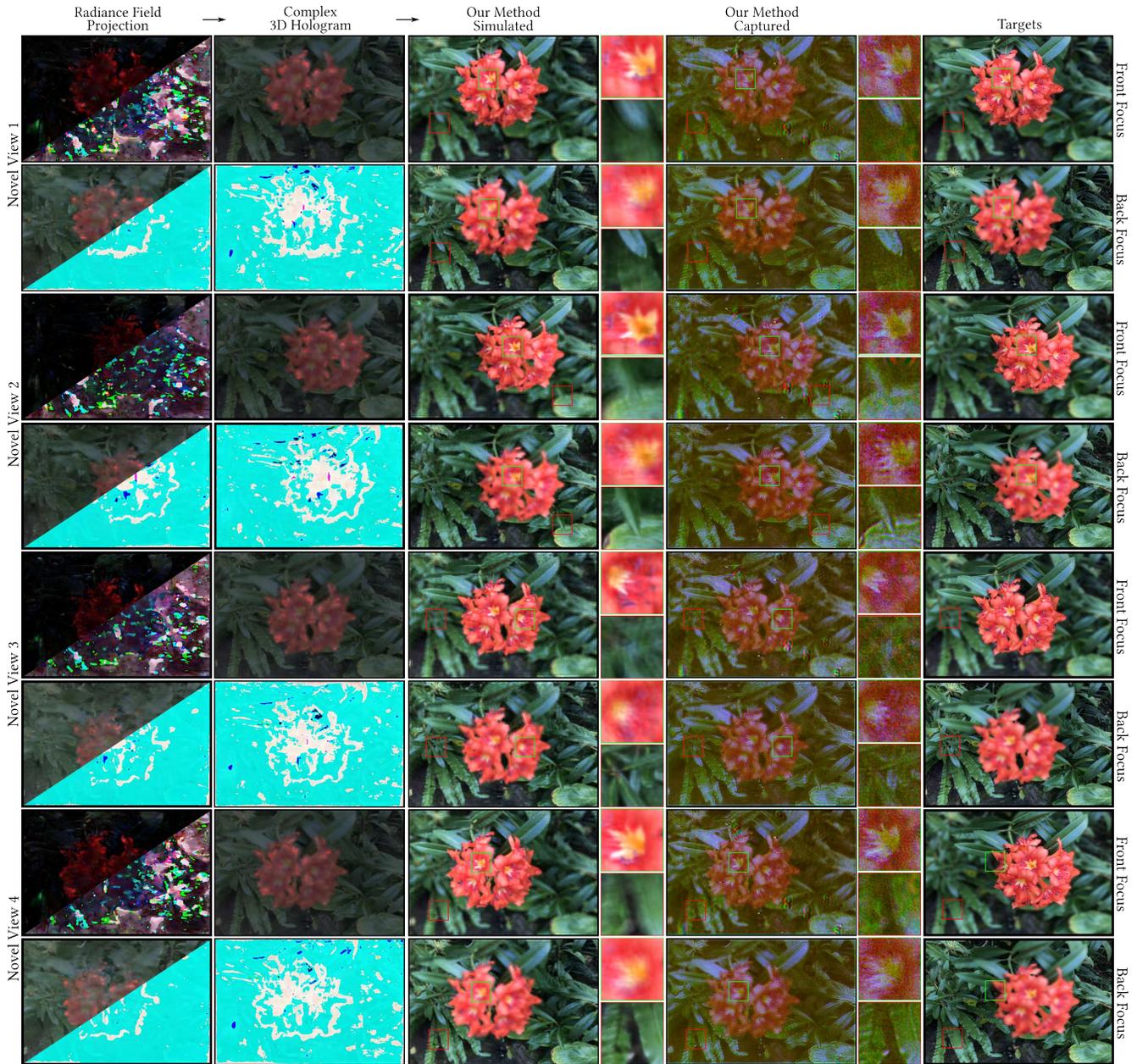


Fig. 12. Extra novel-view comparison of our method on flower scene from the LLFF dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method’s simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

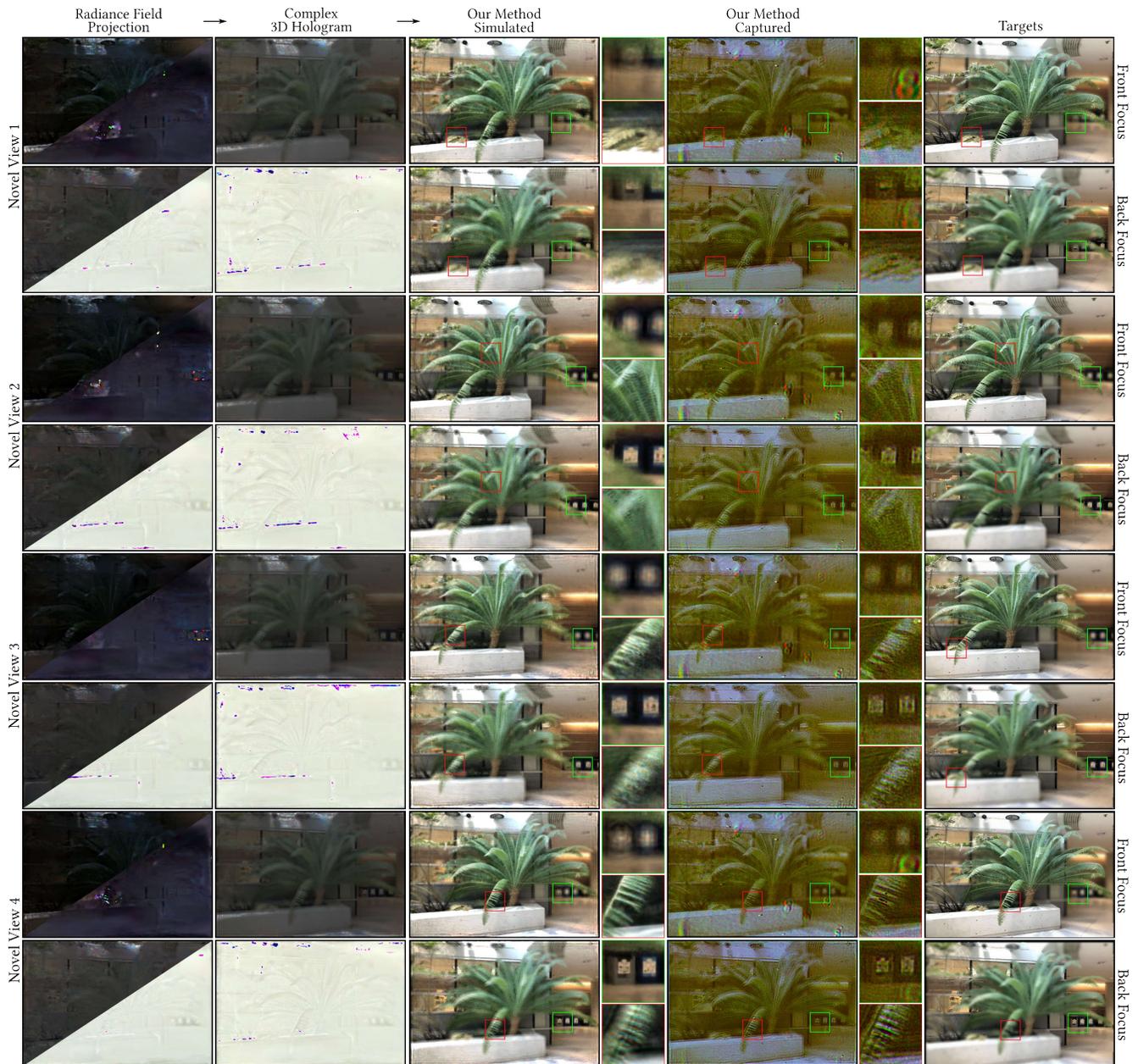


Fig. 13. Extra novel-view comparison of our method on fern scene from the LLFF dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method's simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

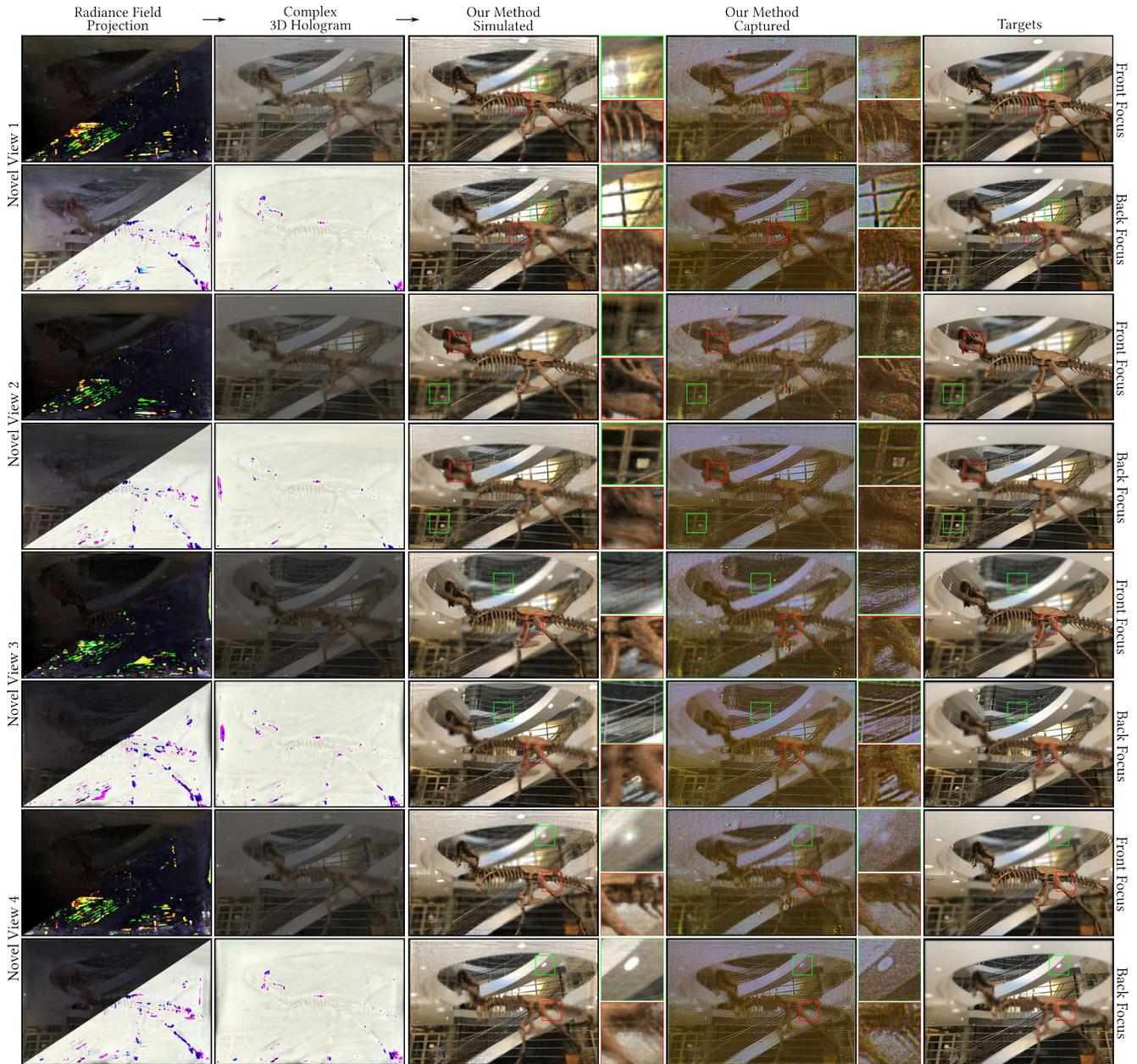


Fig. 14. Extra novel-view comparison of our method on trex scene from the LLFF dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method’s simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.

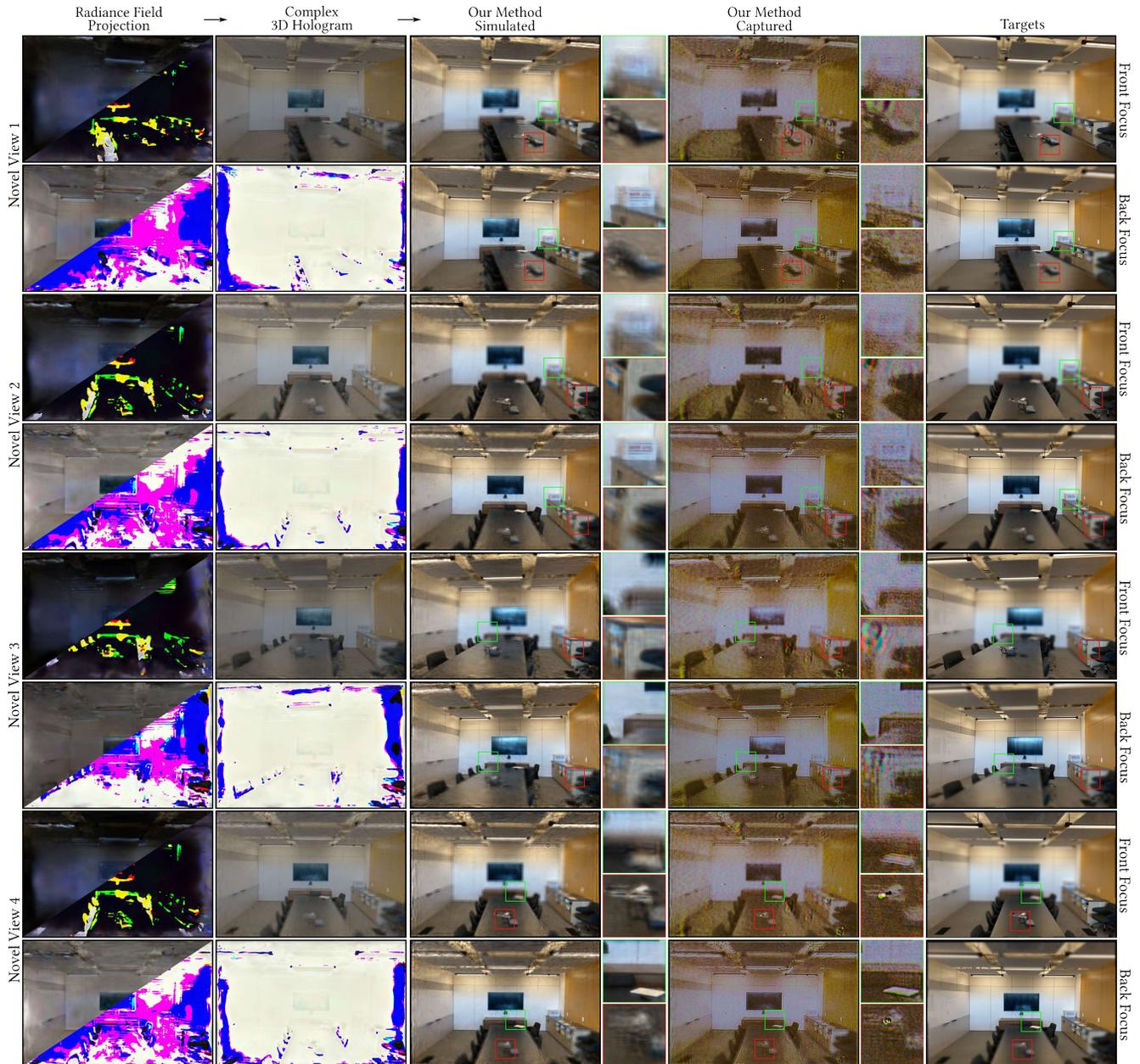


Fig. 15. Extra novel-view comparison of our method on room scene from the LLFF dataset. The first two columns show the radiance field projections and their rendered complex 3D holograms. The central columns present our method's simulated results and experimentally captured results. The rightmost column displays the target images used as optimization objectives in our method.