Deep-Learning-based Phase Retrieval with Time-Multiplexing for Holographic Display

Weiwei Wu (wwwu@stanford.edu)

Abstract—Random-phase Gaussian Wave Splatting (RP-GWS) enables view-dependent holographic rendering by combining Gaussian primitive wavefronts with angular-emission kernels and time multiplexing (TM) [1]. However, in finite-frame settings, spatial fidelity is achieved only approximately and can require many TM frames to converge [1]. In this work, we reformulate primitive-level RP-GWS rendering as a constrained phase-retrieval problem with joint spatial and angular targets. Specifically, we enforce the target Fourier / angular magnitude exactly by constructing the spectrum as $\hat{u}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}e^{\mathrm{i}\phi(\boldsymbol{k})}$, and learn a neural predictor for a strong single-frame phase initialization $\phi_0 = f_\theta(S, V)$. We then apply a lightweight WGS-like TM refinement procedure that improves spatial fidelity while preserving the angular target at every frame. On our Gaussian-primitive benchmark, one learned frame achieves spatial quality comparable to roughly 15–20 RP-GWS frames, while 20-frame refinement yields nearly an order-of-magnitude lower dev-set MSE than the RP-GWS-style baseline. Our results suggest that learned phase retrieval can substantially reduce the frame budget needed for high-quality holographic rendering with exact angular control.

Index Terms—computer-generated holography, Gaussian wave splatting, phase retrieval, time multiplexing, neural holography

1 Introduction

Computer-generated holography (CGH) seeks a phase-only modulation pattern whose propagated optical field reconstructs a desired image or three-dimensional scene. Recent primitive-based approaches have made Gaussian representations particularly attractive because they combine compact scene parameterization with analytic wavefront models [1], [2]. Gaussian Wave Splatting (GWS) renders holograms by converting Gaussian primitives into SLM-plane wave spectra, enabling efficient primitive-based CGH synthesis [2]. However, smooth-phase GWS tends to concentrate energy in the angular spectrum, which underutilizes the SLM bandwidth and limits parallax and natural defocus [1].

Random-phase Gaussian Wave Splatting (RP-GWS) addresses this by injecting random phase and averaging multiple hologram frames through time multiplexing (TM) [1]. While this improves angular emission behavior and view-dependent effects, spatial fidelity is recovered only in expectation across frames, meaning that finite-frame reconstructions may require many TM frames to converge [1]. In this work we reduce this TM burden by reformulating primitive-level RP-GWS rendering as a constrained phase-retrieval problem. Instead of learning both spatial and angular behavior simultaneously, we enforce the angular spectrum exactly and learn only the phase. A neural network predicts a strong single-frame phase initialization conditioned on spatial and angular targets, followed by a lightweight WGS-style TM refinement that improves spatial fidelity while preserving the angular constraint.

Our experiments show that one learned frame already matches the spatial quality of roughly 15–20 RP-GWS frames, and that 20-frame refinement yields nearly an order-of-magnitude reduction in dev-set MSE relative to the RP-GWS-style baseline.

2 Related Work

2.1 Gaussian wave splatting for holography.

Gaussian Wave Splatting introduced an analytic Gaussian-to-hologram transform that maps optimized Gaussian primitives to complex wavefronts on the SLM, together with alpha wave blending for occlusion-aware compositing [2]. The method inherits many strengths of Gaussian scene representations, including compactness and good image quality, but in its smooth-phase form it does not fully exploit the SLM bandwidth for angularly rich holographic rendering [1], [2].

2.2 Random-phase GWS and time multiplexing.

RP-GWS extends GWS by introducing structured random phase in the Fourier domain together with angular-emission kernels and time multiplexing [1]. This formulation substantially improves bandwidth utilization, parallax, and defocus realism. Its main limitation is that spatial fidelity is recovered through frame averaging, so finite-frame reconstructions can require many TM frames before converging to the desired target [1].

2.3 Neural holography and phase retrieval

Learning-based CGH methods have shown that neural networks can predict phase-only holograms more efficiently than purely iterative optimization, particularly when combined with physical propagation models or camera-in-the-loop calibration [3]. Most of these approaches learn a direct mapping from a target image to a phase hologram under a fixed optical configuration.

2.4 Our position

Our method operates in a primitive-level setting where both spatial and angular targets are explicitly specified.

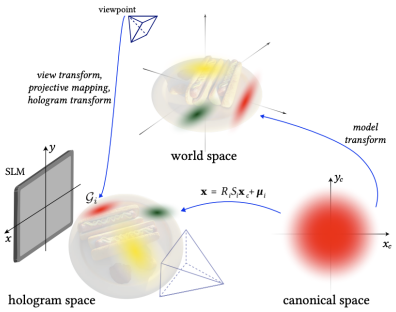

Fig. 1. Gaussian primitive mapping from world space to hologram space in Gaussian Wave Splatting [2].

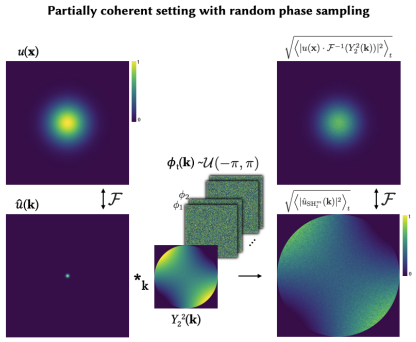

Fig. 2. Random-phase Gaussian Wave Splatting using structured random phase and time multiplexing to achieve view-dependent emission [1].

The inputs to our model are a spatial intensity target $S(\boldsymbol{x})$ and a Fourier / angular magnitude target $V(\boldsymbol{k})$. Here $S(\boldsymbol{x})$ corresponds to the Gaussian spatial footprint of a primitive on the SLM plane, while $V(\boldsymbol{k})$ specifies its desired angular emission profile.

Relative to RP-GWS, we replace random-phase convolution as the main mechanism for satisfying spatial constraints with a learned single-frame phase-retrieval formulation. The network predicts only the phase $\phi(\boldsymbol{k})$, while the Fourier magnitude $\sqrt{V(\boldsymbol{k})}$ is enforced exactly. This formulation makes the method particularly well suited to primitive-level holographic rendering where both spatial and angular structure are prescribed.

3 Theory / Approach

3.1 Gaussian Wave Splatting formulation

Gaussian Wave Splatting converts Gaussian primitives into hologram wave spectra through an analytic Gaussian-to-hologram transform [2]. For a primitive with mean $\boldsymbol{\mu}$, rotation $\boldsymbol{R}$, and scale matrix $\boldsymbol{S}$, the SLM-plane spectrum can be written as

$$\hat{u}(\boldsymbol{k}) = \det(\boldsymbol{J}) \det(\boldsymbol{S})\, \hat{G}(\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{k})\, e^{\mathrm{i}\boldsymbol{k}\boldsymbol{\mu}}, \tag{1}$$

which corresponds to Eq. S24 in the GWS supplement [2]. The spatial footprint of the primitive on the SLM plane is therefore

$$S(\boldsymbol{x}) = \left|\mathcal{F}^{-1}\{\hat{u}(\boldsymbol{k})\}\right|^2. \tag{2}$$

Figure 1 illustrates how Gaussian primitives are mapped from canonical space to hologram space in the GWS rendering pipeline.

While this formulation preserves the Gaussian spatial structure, it does not explicitly control the angular spectrum of the emitted light [1], [2].

3.2 Random-phase Gaussian Wave Splatting

RP-GWS introduces angular control by convolving the primitive spectrum with an angular emission kernel and random phase [1]. At a high level, the spectrum of frame t is

$$\hat{u}^{(t)}_{\mathrm{RP}}(\boldsymbol{k}) = \hat{u}(\boldsymbol{k}) * \left(V(\boldsymbol{k}) e^{\mathrm{i}\phi^{(t)}(\boldsymbol{k})}\right), \qquad t = 1, \ldots, T, \tag{3}$$

which reflects the role of Eq. S40 in the RP-GWS derivation [1]. Time multiplexing then averages multiple hologram frames,

$$S(\boldsymbol{x}) \approx \frac{1}{T} \sum_{t=1}^{T} \left|\mathcal{F}^{-1}\{\hat{u}^{(t)}(\boldsymbol{k})\}\right|^2, \tag{4}$$

so that the spatial footprint is recovered in expectation while the angular emission profile is preserved [1].

Figure 2 shows the RP-GWS pipeline with structured random phase sampling and time multiplexing.

Although this formulation improves parallax and defocus behavior, finite-frame TM can converge slowly.

3.3 Phase-retrieval reformulation

Instead of satisfying spatial and angular constraints through random-phase averaging, we enforce the angular magnitude explicitly and solve only for the phase. Specifically, we construct the angular spectrum as

$$\hat{u}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}\, e^{\mathrm{i}\phi(\boldsymbol{k})}, \tag{5}$$

which guarantees $\left|\hat{u}(\boldsymbol{k})\right|^2 = V(\boldsymbol{k})$ by construction. The remaining problem is therefore to find a phase ϕ such that the reconstructed spatial intensity matches the desired target $S(\boldsymbol{x})$.

3.4 Frame 1: learned phase synthesis

Figure 3 shows the architecture used to generate the first-frame phase. The model takes as input the spatial target $S(\boldsymbol{x})$ and the Fourier / angular target $V(\boldsymbol{k})$. The network also includes a learnable base phase prior $\phi_{\mathrm{base}}(\boldsymbol{k})$. The base phase is first constrained to the valid phase range $(-\pi, \pi)$ through a $\pi \tanh(\cdot)$ operator, then represented by its $\cos(\phi)$ and $\sin(\phi)$ channels. These phase features are concatenated with S and V and passed through a UNet-based backbone (with optional Fourier-operator blocks), which predicts a phase correction $\Delta\phi(\boldsymbol{k})$.

The final first-frame phase is obtained by adding the predicted correction to the base phase:

$$\phi_0(\boldsymbol{k}) = \phi_{\mathrm{base}}(\boldsymbol{k}) + \pi \tanh\big(\Delta\phi(\boldsymbol{k})\big), \tag{6}$$

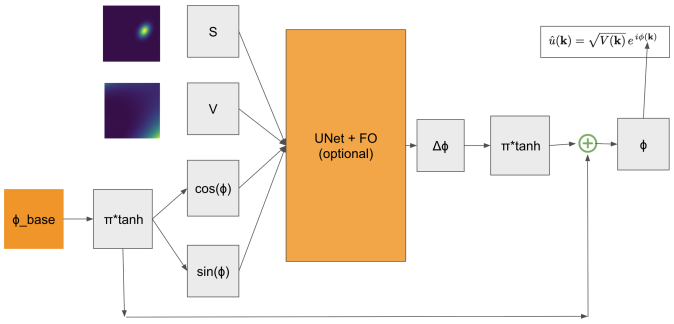

Fig. 3. Model architecture for first-frame phase synthesis. A learnable base phase ϕ_{base} is constrained by $\pi\tanh(\cdot)$ and encoded through $\cos(\phi)$ and $\sin(\phi)$ features. Together with the spatial target S and Fourier target V, these inputs are passed to a UNet-based network that predicts a phase correction $\Delta\phi$. The corrected phase ϕ is then used to construct the constrained angular spectrum $\hat{u}(k) = \sqrt{V(k)}e^{i\phi(k)}$.

where $\phi_0(\boldsymbol{k})$ denotes the predicted phase used for the first hologram frame.

Using this phase, we construct the first-frame angular spectrum

$$\hat{u}^{(1)}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}\, e^{\mathrm{i}\phi_0(\boldsymbol{k})}, \tag{7}$$

where $\hat{u}^{(1)}(\boldsymbol{k})$ denotes the complex spectrum of the first hologram frame. The factor $\sqrt{V(\boldsymbol{k})}$ enforces the desired Fourier / angular magnitude, while the phase term $e^{\mathrm{i}\phi_0(\boldsymbol{k})}$ provides the degrees of freedom needed to match the spatial target.

The corresponding spatial-domain field and reconstruction are

$$u^{(1)}(\boldsymbol{x}) = \mathcal{F}^{-1}\{\hat{u}^{(1)}(\boldsymbol{k})\}(\boldsymbol{x}), \tag{8}$$

$$S^{(1)}(\boldsymbol{x}) = \left|u^{(1)}(\boldsymbol{x})\right|^2, \tag{9}$$

where $\mathcal{F}^{-1}\{\cdot\}$ denotes the inverse Fourier transform and $u^{(1)}(\boldsymbol{x})$ is the complex optical field in the spatial domain.

The network is trained with a spatial reconstruction loss

$$\mathcal{L}_{\text{spatial}} = \|S^{(1)} - S\|_2^2, \tag{10}$$

i.e., a mean squared error (MSE) loss between the reconstructed spatial intensity $S^{(1)}(\boldsymbol{x})$ and the target $S(\boldsymbol{x})$. This learned first frame already provides a strong approximation to the spatial target while satisfying the angular constraint exactly.

3.5 Frames 2–N: lightweight TM generation

After the first frame is predicted by the network, additional time-multiplexed frames are generated without further neural inference. For each frame $t \geq 2$, we start from the learned phase and apply a small perturbation

$$\tilde{\phi}^{(t)}(\boldsymbol{k}) = \mathrm{wrap}\Big(\phi_0(\boldsymbol{k}) + \epsilon_\phi\, \eta^{(t)}(\boldsymbol{k})\Big), \tag{11}$$

where $\tilde{\phi}^{(t)}(\boldsymbol{k})$ denotes the phase perturbation used to generate the t-th TM frame, ϵ_ϕ controls the perturbation magnitude, $\eta^{(t)}(\boldsymbol{k})$ is random noise, and $\mathrm{wrap}(\cdot)$ denotes phase wrapping that keeps the phase within the interval $(-\pi, \pi)$.

The corresponding spectrum and spatial field are

$$\hat{u}^{(t)}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}\, e^{\mathrm{i}\tilde{\phi}^{(t)}(\boldsymbol{k})}, \tag{12}$$

$$u^{(t)}(\boldsymbol{x}) = \mathcal{F}^{-1}\{\hat{u}^{(t)}(\boldsymbol{k})\}(\boldsymbol{x}). \tag{13}$$

To improve spatial fidelity across TM frames, we introduce a per-frame spatial target $S^{(t)}_{\text{tgt}}(\boldsymbol{x})$, which guides each new frame toward correcting the residual error of the current running reconstruction. The precise update rule for this target is described in the next subsection.

Given $S^{(t)}_{\text{tgt}}(\boldsymbol{x})$, we perform a single alternating-projection step similar to weighted Gerchberg–Saxton:

$$u^{(t)}_{\text{proj}}(\boldsymbol{x}) = \sqrt{S^{(t)}_{\text{tgt}}(\boldsymbol{x})}\, e^{\mathrm{i}\angle u^{(t)}(\boldsymbol{x})}, \tag{14}$$

$$\hat{u}^{(t)}_{\text{proj}}(\boldsymbol{k}) = \mathcal{F}\{u^{(t)}_{\text{proj}}(\boldsymbol{x})\}, \tag{15}$$

$$\phi^{(t)}(\boldsymbol{k}) = \angle\hat{u}^{(t)}_{\text{proj}}(\boldsymbol{k}), \tag{16}$$

where $\angle(\cdot)$ extracts the phase of a complex number and $\mathcal{F}\{\cdot\}$ denotes the Fourier transform.

Finally, the Fourier magnitude constraint is re-imposed

$$\hat{u}^{(t)}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}\, e^{\mathrm{i}\phi^{(t)}(\boldsymbol{k})}, \tag{17}$$

ensuring that the angular spectrum remains exact for every frame.

3.6 Residual-steered spatial targets

To reduce systematic bias in the TM reconstruction, we adapt the spatial target using the running reconstruction residual. We initialize the running-average reconstruction with the model-predicted first frame,

$$\bar{S}^{(1)}(\boldsymbol{x}) = |u^{(1)}(\boldsymbol{x})|^2. \tag{18}$$

For a running average after t frames,

$$\bar{S}^{(t)}(\boldsymbol{x}) = \frac{1}{t}\sum_{\tau=1}^{t} |u^{(\tau)}(\boldsymbol{x})|^2, \tag{19}$$

the spatial residual is

$$R^{(t)}(\boldsymbol{x}) = S(\boldsymbol{x}) - \bar{S}^{(t)}(\boldsymbol{x}). \tag{20}$$

The target used for the next TM frame is then

$$S^{(t+1)}_{\text{tgt}}(\boldsymbol{x}) = \Pi_{\geq 0}\Big(S(\boldsymbol{x}) + \gamma R^{(t)}(\boldsymbol{x})\Big), \tag{21}$$

where $\Pi_{\geq 0}(\cdot)$ denotes projection onto non-negative intensities and γ controls the correction strength.

Thus, the first TM refinement frame ($t = 2$) uses a target derived from the residual after the learned first frame, and subsequent frames continue to update this target using the running-average error.

4 Analysis and Results

We evaluate the proposed method on a primitive-level benchmark in which each sample is defined by a target spatial intensity $S(\boldsymbol{x})$ and a target Fourier / angular magnitude $V(\boldsymbol{k})$ for a single Gaussian primitive. We compare against a naive RP-GWS-style baseline that uses the convolution heuristic and spatial-domain time-multiplexed averaging. Our primary quantitative metric is dev-set MSE between

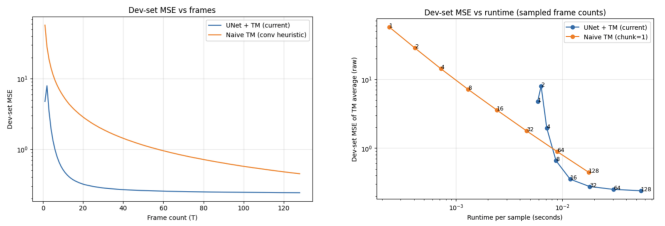

Fig. 4. Quantitative evaluation. **Left:** dev-set MSE versus frame count. Our method converges substantially faster and reaches a lower final error than the RP-GWS-style baseline. **Right:** dev-set MSE versus runtime per sample (batch size 1). Although the learned first frame adds inference cost, our method achieves better reconstruction quality at comparable runtime.

the averaged spatial reconstruction and the target $S(\boldsymbol{x})$. We also report runtime per sample in our implementation with batch size 1.

Figure 4 summarizes the quantitative behavior. The left plot shows dev-set MSE as a function of frame count. Across the full range of sampled frame budgets, our learned initialization plus lightweight TM refinement converges substantially faster and reaches a lower final error than the baseline. The gain is most pronounced in the practically important low-frame regime: within roughly the first 20 frames, our method reduces dev-set MSE by nearly an order of magnitude relative to the RP-GWS-style baseline. After this early regime, both methods continue to improve more gradually, but our method maintains a consistent advantage even at 128 frames.

The right plot in Figure 4 shows the same comparison against runtime per sample. Because our method includes a learned first-frame prediction, it introduces a fixed inference overhead that is visible at very small frame counts. However, once a modest number of refinement frames are used, this overhead becomes amortized and our method achieves a substantially better quality–runtime tradeoff. In practice, the method operates in the millisecond regime per sample while maintaining significantly lower dev-set MSE than the baseline at comparable runtime.

We next examine the qualitative behavior of the two approaches. The main questions are: (1) how much spatial quality is obtained from the learned first frame alone, (2) how quickly refinement improves the spatial reconstruction, and (3) whether the angular target remains accurate throughout.

Figure 5 compares the target pair (S, V), the RP-GWS baseline at 20 frames, and our learned first frame before TM refinement. Qualitatively, a single predicted frame already captures the correct spatial support and approximate Gaussian structure, even though it still contains visible noise and bias. This is important because the baseline requires many averaged random-phase frames to reach a comparable spatial footprint. At the same time, the angular target in our method is exact by construction: the stored V_{target} matches the desired Fourier magnitude, whereas the baseline exhibits clear angular mismatch in V_{pred}. This figure therefore highlights the role of the learned phase predictor: even before refinement, it provides a strong and physically consistent initialization.

Figure 6 shows the equal-frame comparison at $T = 20$.

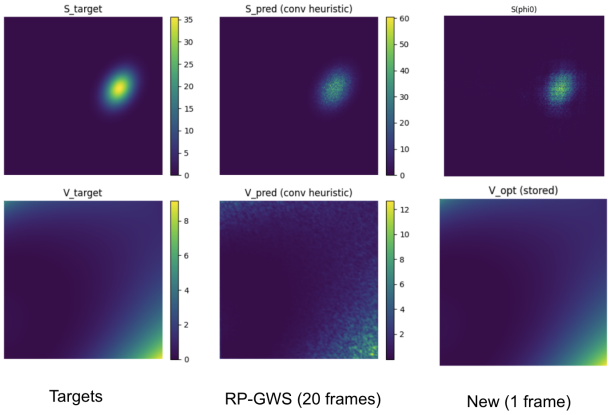

Fig. 5. One-frame comparison. Left: target spatial and angular quantities $(S_{\text{target}}, V_{\text{target}})$. Middle: RP-GWS-style baseline after 20 frames. Right: our learned first frame. Even before TM refinement, the learned phase produces the correct coarse spatial footprint while preserving the angular target exactly by construction.

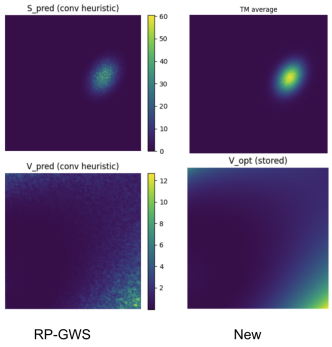

Fig. 6. Comparison at 20 frames. The proposed method produces a cleaner TM-averaged spatial reconstruction than the RP-GWS-style baseline while maintaining the exact target angular magnitude.

Here the benefit of the proposed refinement procedure becomes clear. The TM-averaged output from our method is smooth, correctly localized, and much closer to the Gaussian target than the baseline result, which still contains visible speckle-like fluctuations in both domains. More importantly, the Fourier magnitude remains exact in our method because each frame is explicitly reconstructed as $\hat{u}^{(t)}(\boldsymbol{k}) = \sqrt{V(\boldsymbol{k})}e^{\mathrm{i}\phi^{(t)}(\boldsymbol{k})}$. In contrast, the RP-GWS baseline continues to exhibit a noisy angular pattern. This observation is consistent with the quantitative curves in Figure 4: around 20 frames, our method already enters a low-error regime while the baseline remains far behind.

Figure 7 shows the long-horizon comparison at $T = 128$. By this point the baseline has improved substantially compared with its low-frame behavior, but it still remains noisier spatially and continues to deviate from the desired angular target. The gap is smaller than in the 20-frame regime, which is also reflected in the flattening of both curves in Figure 4, but the advantage of the proposed method persists. This behavior suggests that the main benefit of our method is not only a better final asymptote, but more importantly a much faster path to high-quality holograms.

Taken together, these results support two key observations. First, the learned first-frame phase already captures

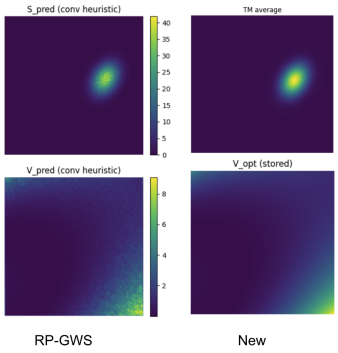

Fig. 7. Comparison at 128 frames. The baseline improves with longer averaging, but our method still produces a cleaner spatial reconstruction and continues to satisfy the angular target exactly.

much of the desired spatial structure and provides a better starting point than random-phase averaging alone. Second, once the angular magnitude is treated as a hard constraint, a lightweight WGS-like refinement procedure is sufficient to obtain high spatial fidelity with significantly fewer frames than the RP-GWS baseline.

5 Discussion and Conclusion

We presented a primitive-level phase-retrieval formulation for holographic rendering that replaces RP-GWS-style random-phase averaging with a learned single-frame phase predictor and lightweight TM refinement. The key design choice is to enforce the Fourier / angular magnitude exactly at every frame and optimize only through the angular domain phase. This leads to improved low-frame performance and faster convergence than a naive RP-GWS TM baseline.

As a next step, we plan to integrate the proposed per-primitive phase-retrieval formulation into full scene rendering pipelines. Real Gaussian scenes typically contain on the order of 5×10^5 primitives, making it important to evaluate how the learned first-frame initialization and TM refinement scale in realistic settings with Alpha Wave Blending. Future work will therefore focus on scene-level rendering experiments and quantitative evaluation using standard image-quality metrics such as PSNR and MSE.

References

[1] B. Chao, J. Yang, S. Choi, M. Gopakumar, R. Koiso, and G. Wetzstein, "Random-phase gaussian wave splatting for computer-generated holography," *arXiv preprint arXiv:2508.17480*, 2025.

[2] S. Choi, B. Chao, J. Yang, M. Gopakumar, and G. Wetzstein, "Gaussian wave splatting for computer-generated holography," *ACM Transactions on Graphics*, vol. 44, no. 4, pp. 1–13, 2025.

[3] Y. Peng, S. Choi, N. Padmanaban, and G. Wetzstein, "Neural holography with camera-in-the-loop training," *ACM Transactions on Graphics*, vol. 39, no. 6, pp. 185:1–185:14, 2020.