MPP-GS: MULTI-PROJECTION PACKING FOR HIGH-FIDELITY 3DGS WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermarking 3D Gaussian Splatting (3DGS) assets requires modifying selected Gaussian primitives. When watermark carriers are concentrated in a few spatial regions, those regions must carry more of the embedding burden, potentially requiring stronger local perturbations that degrade rendering fidelity. However, existing methods often lack explicit mechanisms for spatially balanced carrier allocation. In this paper, we introduce MPP-GS, a high-fidelity 3DGS watermarking framework with two stages. In carrier construction, Multi-Projection Packing (MPP) organizes Gaussians on a UV grid and assigns each Gaussian multiple candidate destinations, reducing congestion and overflow while producing a more balanced carrier. In watermark embedding, Complementary Pair Modulation (CPM) exploits this carrier to distribute parent–auxiliary pairs and coordinate their updates, reducing rendering interference while preserving decodability. Across three benchmarks (Blender, LLFF, and Mip-NeRF 360), MPP reduces average carrier overflow from 17.75% to 0.08%, retaining substantially more Gaussian primitives. Together, balanced carrier construction and low-interference watermark embedding enable MPP-GS to achieve higher rendering fidelity with comparable or better decoding accuracy across multiple payloads. At 64 bits, MPP-GS achieves 35.15 dB PSNR and 95.95% bit accuracy, compared with 32.79 dB and 91.17% for W2M. Code will be made publicly available on GitHub.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) provides an explicit 3D representation whose assets are readily edited and redistributed (Kerbl et al., 2023), motivating watermarking for ownership protection. Embedding recoverable messages with minimal rendering distortion requires careful allocation of watermark updates across Gaussians. Existing methods primarily select suitable carriers and control their updates (Huang et al., 2024b; Chen et al., 2025; Jang et al., 2025; Li et al., 2026), with recent work further considering carrier budgets and coverage (Cai et al., 2026). However, these strategies do not explicitly control the spatial distribution of selected carriers. If many carriers are concentrated in a few regions, those regions must carry more of the embedding burden, potentially requiring larger local perturbations and causing greater rendering distortion. This suggests that, beyond selecting suitable carriers, their spatial distribution should also be carefully organized. We therefore ask: *can organizing a pretrained 3DGS into explicit allocation units enable more balanced watermark allocation and thereby improve rendering fidelity at comparable decoding accuracy?*

A natural way to control carrier distribution is to organize 3D Gaussian primitives in a structured domain. UV parameterization provides such a mechanism by mapping otherwise unordered Gaussians onto a regular 2D grid, where each location serves as an explicit allocation unit. We therefore use analytical UV parameterization (Rai et al., 2025) to organize pretrained Gaussians into a multi-layer grid with K slots per location. This representation makes local occupancy and capacity explicit, allowing watermarking opportunities to be distributed across the grid through location-wise quotas before selecting primitives by importance within each location. Explicit slot-to-primitive correspondence then links these selections back to the native 3DGS. In this way, UV parameterization provides a structured basis for controlling and balancing carrier allocation across locations.

Building on this structured UV representation, a key limitation is how Gaussians are mapped onto the grid. In vanilla single-projection mapping, each Gaussian is assigned only one prescribed UV destination. As illustrated in Figure 1(a), multiple Gaussians may therefore collide at the same location, with no alternative destinations available to relieve local congestion. Since each location

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

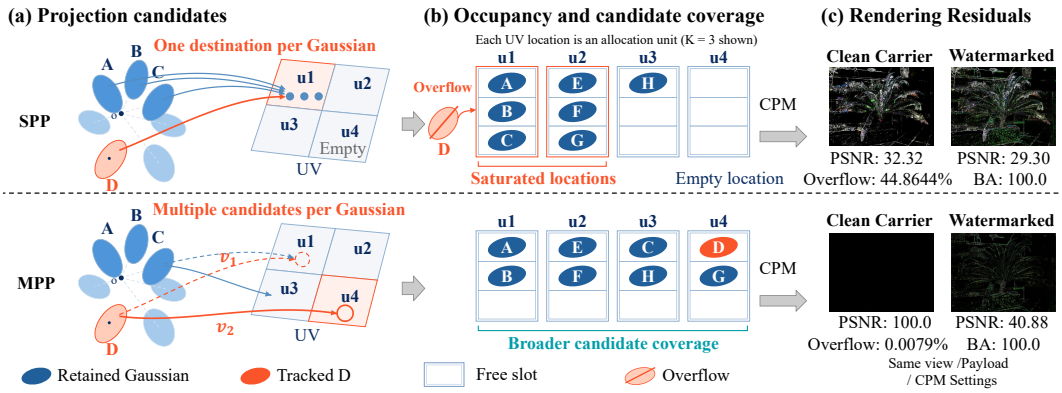


Figure 1: **Single-projection packing (SPP) vs. multi-projection packing (MPP).** (a) SPP can cause congestion and overflow, while multiple candidates enable alternative placements. (b) MPP broadens UV coverage and reduces saturation. (c) MPP lowers overflow and improves rendering fidelity at comparable bit accuracy.

contains only K slots, Gaussians beyond this capacity are treated as overflow even when unused slots remain elsewhere. This can produce highly uneven occupancy, with saturated locations coexisting with sparsely occupied or empty ones (Figure 1(b)). Such overflow removes potentially useful watermark carriers and can degrade reconstruction fidelity, while concentrated occupancy restricts how broadly watermarking opportunities can be distributed across the UV domain (Figure 1(c)). These limitations motivate a more flexible packing strategy that exploits alternative UV placements to reduce overflow and broaden candidate support for balanced watermark allocation.

To this end, we introduce **Multi-Projection Packing (MPP)** to reduce overflow and broaden candidate support for watermark allocation. Multi-Camera Projection provides each Gaussian with alternative UV locations, while occupancy-aware assignment favors less crowded destinations. Importance-aware packing further retains important Gaussians and rescues overflowed primitives through alternative placements, reducing local saturation and expanding effective UV coverage. Building on this carrier, **Complementary Pair Modulation (CPM)** distributes parent–auxiliary pairs across feasible UV locations using MPP's occupancy and coordinates their updates through complementary parameterization. It further reuses MPP's projection directions for nullspace modulation to reduce rendering interference. Across three benchmarks and multiple payloads, MPP-GS achieves higher rendering fidelity with comparable or better decoding accuracy. Comparisons using the same CPM design across alternative carriers further support the benefit of MPP's carrier construction. Our main contributions are summarized as follows:

- We introduce MPP-GS, a high-fidelity 3DGS watermarking framework that separates carrier construction from watermark embedding to support spatially balanced watermark allocation.
- We propose Multi-Projection Packing (MPP), which provides alternative UV placements to reduce overflow and expand effective carrier coverage under a fixed capacity, enabling broader and more balanced candidate support.
- We develop Complementary Pair Modulation (CPM), which exploits MPP's carrier occupancy and projection directions to distribute watermark pairs across locations and reduce rendering interference.

2 Related Work

3DGS Watermarking. Earlier 3D watermarking spans meshes and point clouds (Ohbuchi et al., 2002; Praun et al., 1999; Yoo et al., 2022) and neural radiance fields (Luo et al., 2023; Jang et al., 2024; Li et al., 2023; Song et al., 2024; Huang et al., 2024a; Luo et al., 2025). The growing use of 3DGS (Kerbl et al., 2023) in virtual reality (Qiu et al., 2025; Zhan et al., 2025; Dong et al., 2025), scene reconstruction (Yuan et al., 2025; Zhao et al., 2025; Zhu et al., 2024), and generation (Yi et al., 2024; Kong et al., 2025; Ju & Li, 2025) has motivated dedicated watermarking methods (Zhang et al., 2024b; Huang et al., 2024b; Chen et al., 2025; Li et al., 2025; Huang et al., 2025). Representative methods improve carrier selection, expansion, and parameter updates through

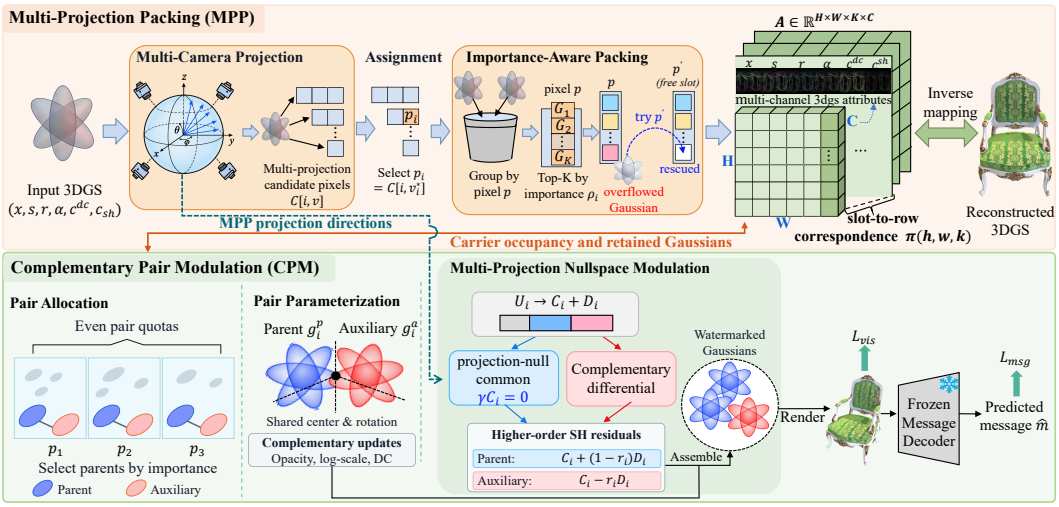


Figure 2: **Overview of MPP-GS. Top:** MPP constructs the UV carrier through *Multi-Camera Projection*, *Occupancy-Aware Assignment*, and *Importance-Aware Packing*, reducing overflow and expanding effective UV coverage while preserving explicit slot-to-primitive correspondence. **Bottom:** CPM embeds the watermark through *Pair Allocation*, *Pair Parameterization*, and *Nullspace Modulation*. It distributes parent–auxiliary pairs across UV locations and coordinates their updates to reduce rendering interference.

uncertainty-aware perturbations (Huang et al., 2024b), frequency-guided densification (Jang et al., 2025), and learnable SH modulation (Chen et al., 2025). More recent work further refines watermark allocation. Where, What, Why (Cai et al., 2026) uses per-Gaussian priors to select and expand message carriers with separate visual compensators, while W2M (Li et al., 2026) controls writing locations and strengths through policy-guided allocation. However, improved carrier selection and update control do not necessarily ensure that writing opportunities are spatially balanced. We complement these approaches by organizing pretrained Gaussians into explicit allocation units and using location-wise quotas to support a more balanced carrier.

Structured 3DGS. Structured 3DGS representations arrange Gaussian primitives in regular domains. Voxel-grid (Zhang et al., 2024a) and image-like (Szymanowicz et al., 2024; Huang et al., 2025; Rai et al., 2026) representations have been explored for reconstruction, generation, and storage. However, discretization, representation-specific fitting, or learned prediction can complicate their direct use as watermark carriers while preserving explicit correspondence to pretrained Gaussian primitives. Analytical UV parameterization instead reorganizes pretrained Gaussians into a multi-layer grid with addressable units for location-wise allocation. UVGS (Rai et al., 2025) constructs such a representation through spherical projection, assigning each Gaussian a single prescribed destination. This single-destination constraint can cause overflow and concentrate retained candidates at congested locations even when free capacity remains elsewhere. MPP addresses this limitation by introducing alternative projection candidates to reduce overflow and expand effective UV coverage while preserving explicit slot-to-primitive correspondence for watermark allocation.

3 Method

As shown in Figure 2, MPP-GS consists of two stages. First, **MPP** constructs a balanced UV carrier through *Multi-Camera Projection*, *Occupancy-Aware Assignment*, and *Importance-Aware Packing*. Second, **CPM** embeds the watermark on this carrier through *Pair Allocation*, *Pair Parameterization*, and *Nullspace Modulation*. Lastly, the watermark parameters are optimized for visual rendering fidelity and watermark message recovery objectives.

3.1 Multi-Projection Packing

Given a pretrained 3DGS asset $\mathcal{G} = \{g_i\}_{i=1}^{N}$, MPP constructs a UV carrier $\mathbf{A} \in \mathbb{R}^{H\times W\times K\times C}$ with K slots per location and a total capacity of $H \times W \times K$ slots. Each UV location serves as an

allocation unit, and its retained Gaussians form candidate carriers for watermark embedding. Each occupied slot stores the complete attribute vector of one Gaussian:

$$\mathbf{a}_i = \left[\mathbf{x}_i,\, \mathbf{s}_i,\, \mathbf{r}_i,\, \tilde{\alpha}_i,\, \mathbf{c}_i^{dc},\, \mathbf{c}_i^{sh}\right] \in \mathbb{R}^C, \tag{1}$$

where the entries denote center, log-scale, rotation quaternion, logit opacity, DC color, and higher-order spherical-harmonic (SH) coefficients, respectively. With SH degree $\ell_{\max}$, $L = (\ell_{\max}+1)^2$ and $C = 14 + 3(L-1)$. Thus, MPP preserves the native Gaussian parameters while determining how retained primitives are arranged within the carrier.

Multi-Camera Projection. A single projection assigns each Gaussian only one UV destination, preventing it from using free capacity elsewhere when that destination is congested. MPP instead provides each Gaussian with multiple candidate UV locations. Given the training camera centers $\{\mathbf{o}_j\}_{j=1}^M$ and a robust scene center $\bar{\mathbf{x}}$, we construct V projection directions from representative camera directions together with additional directions that improve spatial coverage. For Blender and Mip-NeRF 360, we combine farthest-point-sampled camera directions with ring directions around the mean viewing direction; for forward-facing LLFF scenes, we use ring directions (see details in Appendix A.1). For each direction v, we construct a local frame $\mathbf{R}_v \in \mathbb{R}^{3\times3}$ and transform the centered Gaussian position as

$$\tilde{\mathbf{p}}_i^{(v)} = \mathbf{R}_v\,(\mathbf{x}_i - \bar{\mathbf{x}})\,, \qquad \mathbf{C}[i,v] = \mathrm{Proj}\left(\tilde{\mathbf{p}}_i^{(v)}; H, W\right), \tag{2}$$

where $\mathbf{C} \in \mathbb{Z}^{N\times V}$ stores the candidate UV locations and $\mathrm{Proj}(\cdot; H, W)$ denotes equirectangular spherical discretization onto the $H \times W$ grid. Each Gaussian therefore obtains V candidate destinations without increasing carrier capacity.

Occupancy-Aware Assignment. We select one candidate UV location for each Gaussian according to the current occupancy. Let v_i^* denote the selected projection direction for Gaussian i, and let $\mathrm{occ}[p]$ denote the number of Gaussians currently assigned to location p. Starting from an initial assignment, we iteratively update

$$v_i^* \leftarrow \arg\min_{v\in\{1,\dots,V\}} \left(\mathrm{occ}[\mathbf{C}[i,v]] + \epsilon_{i,v}\right), \tag{3}$$

where $\epsilon_{i,v}$ is a small random perturbation used only to break occupancy ties. The selected location is $p_i = \mathbf{C}[i, v_i^*]$. This assignment favors less crowded candidate destinations before packing, thereby reducing local congestion within the fixed $H \times W \times K$ carrier.

Importance-Aware Packing. Packing enforces the local capacity constraint when more than K Gaussians are assigned to the same UV location. For location p, let $\mathcal{S}_p = \{\, i \mid p_i = p \,\}$. If $|\mathcal{S}_p| \le K$, all assigned Gaussians are retained. Otherwise, we provisionally retain the K Gaussians with the highest importance scores

$$\rho_i = \sigma(\tilde{\alpha}_i)\sqrt{\prod_{d=1}^{3}\exp(s_{i,d})}, \tag{4}$$

which combine opacity with a square-root volume proxy. The remaining Gaussians are marked as overflow. To recover overflowed primitives, we revisit their alternative candidate locations. Let $\mathcal{P}_i = \{\mathbf{C}[i,v]\}_{v=1}^V$ denote the distinct candidate locations of overflowed Gaussian i. Processing overflowed Gaussians in descending order of ρ_i, we define the score of candidate slot (p,k) as

$$q(p,k) = \begin{cases} -\infty, & \text{if slot } (p,k) \text{ is empty,} \\ \rho_j, & \text{if slot } (p,k) \text{ is occupied by Gaussian } j. \end{cases} \tag{5}$$

We select the weakest accessible slot:

$$(p_i^\dagger, k_i^\dagger) = \arg\min_{\substack{p\in\mathcal{P}_i \\ k\in\{1,\dots,K\}}} q(p,k). \tag{6}$$

Gaussian i is inserted if $\rho_i > q(p_i^\dagger, k_i^\dagger) + \delta$, where $\delta \ge 0$ is a replacement margin. An empty slot directly rescues the Gaussian, whereas an occupied slot is replaced only when its current occupant has sufficiently lower importance. Overflow rescue therefore improves primitive retention without

increasing the carrier budget. After rescue, let $\mathcal{R}$ denote the retained Gaussian set and $N' = |\mathcal{R}| \leq N$. We define the slot-to-primitive correspondence

$$\pi(h, w, k) \in \{-1\} \cup \{0, \ldots, N' - 1\}, \tag{7}$$

where $\pi(h, w, k) = -1$ denotes an empty slot and nonnegative values index the compact retained Gaussian set. Each occupied slot satisfies $\mathbf{A}[h, w, k, :] = \mathbf{a}_{\pi(h,w,k)}$. During implementation, we store only the occupied attribute rows and their slot indices, rather than materializing the dense carrier. The resulting occupancy specifies the retained candidates and free slots available at each UV location for subsequent pair allocation.

3.2 Complementary Pair Modulation

Given the UV carrier constructed by MPP, CPM embeds the watermark through three steps: *Pair Allocation*, *Pair Parameterization*, and *Nullspace Modulation*. Pair Allocation distributes writing opportunities across UV locations, while the latter two coordinate parent–auxiliary updates to reduce rendering interference.

Pair Allocation. CPM first uses MPP's carrier occupancy to distribute parent–auxiliary pairs across UV locations. Each auxiliary occupies an unused slot at its parent's location, with at most one auxiliary per parent. Let n_p denote the number of slots occupied by retained Gaussians at location p before pair allocation. Since each pair requires both a retained parent and a free slot, the local pair capacity is

$$c_p = \min(n_p, K - n_p). \tag{8}$$

We allocate $B_{\text{aux}} = \min(N_{\text{occ}}, \sum_p c_p)$ auxiliaries, where $N_{\text{occ}} = |\{p : n_p > 0\}|$ is the number of occupied locations. The number of pairs assigned to each location defines its *writing quota*. We distribute this budget as evenly as possible across locations with positive pair capacity, subject to the local bound c_p. Within each location, parents are selected in descending order of importance ρ_i until the quota is filled. Using the slot-to-primitive correspondence π, each selected parent is assigned a distinct empty slot for its auxiliary. The auxiliary shares the parent's 3D center and rotation, and both remain standard primitives compatible with the original 3DGS renderer. Further allocation details are provided in Appendix B.1.

Pair Parameterization. For an allocated parent–auxiliary pair i, let $\alpha_i = \sigma(\tilde{\alpha}_i)$ denote the original opacity and $r_i \in (0, 1)$ a learnable mixing fraction. We split the opacity between the parent and auxiliary as

$$\alpha_i^p = r_i \alpha_i, \qquad \alpha_i^a = \frac{(1 - r_i)\alpha_i}{1 - r_i \alpha_i}, \qquad 1 - (1 - \alpha_i^p)(1 - \alpha_i^a) = \alpha_i. \tag{9}$$

Thus, the pair preserves the original opacity under alpha composition. We use the same mixing weights to coordinate appearance updates. Let $\mathbf{s}_i$ denote the original log-scale and $\mathbf{f}_i^{dc} = \mathbf{c}_i^{dc}$ the DC feature. Given trainable residuals $\Delta\mathbf{s}_i$ and $\Delta\mathbf{f}_i^{dc}$, we parameterize

$$\begin{aligned} \mathbf{s}_i^p &= \mathbf{s}_i + \Delta\mathbf{s}_i, & \mathbf{s}_i^a &= \mathbf{s}_i - \Delta\mathbf{s}_i, \\ \mathbf{f}_i^{dc,p} &= \mathbf{f}_i^{dc} + (1 - r_i)\Delta\mathbf{f}_i^{dc}, & \mathbf{f}_i^{dc,a} &= \mathbf{f}_i^{dc} - r_i \Delta\mathbf{f}_i^{dc}. \end{aligned} \tag{10}$$

The scale perturbation is symmetric in log space, while the DC updates satisfy $r_i \mathbf{f}_i^{dc,p} + (1 - r_i)\mathbf{f}_i^{dc,a} = \mathbf{f}_i^{dc}$, so the weighted DC residual cancels. These relations hold at the parameter level, because projected footprints and visibility also affect the rendered response, and visual fidelity is additionally enforced through the rendering loss. Higher-order SH updates do not fully cancel under the same pairwise weighting. We therefore further constrain their directional response using the projection directions introduced by MPP.

Nullspace Modulation. Let $\{\mathbf{d}_v\}_{v=1}^V$ denote the viewing directions corresponding to the V MPP projection directions, expressed in the renderer's direction convention. For each direction, $\mathbf{y}(\mathbf{d}_v) \in \mathbb{R}^{L-1}$ denotes the higher-order SH basis. Stacking these basis vectors gives $\mathbf{Y} \in \mathbb{R}^{V \times (L-1)}$. We use the nullspace projector

$$\mathbf{P}_\perp = \mathbf{I} - \mathbf{Y}^\dagger \mathbf{Y}, \qquad \mathbf{C}_i = \mathbf{P}_\perp \mathbf{U}_i, \qquad \mathbf{D}_i = (\mathbf{I} - \mathbf{P}_\perp)\mathbf{U}_i, \tag{11}$$

where $\mathbf{Y}^\dagger$ is the Moore–Penrose pseudoinverse and $\mathbf{U}_i \in \mathbb{R}^{(L-1)\times 3}$ is the trainable higher-order SH residual of pair i. By construction, $\mathbf{Y}\mathbf{C}_i = \mathbf{0}$, so $\mathbf{C}_i$ produces zero SH response along the representative MPP directions. A nonzero residual in this nullspace remains possible whenever $\mathrm{rank}(\mathbf{Y}) < L-1$. Let $\mathbf{F}_i^{sh} \in \mathbb{R}^{(L-1)\times 3}$ denote the original higher-order SH coefficients. We assign the decomposed residuals to the pair as

$$\mathbf{F}_i^{sh,p} = \mathbf{F}_i^{sh} + \mathbf{C}_i + (1-r_i)\mathbf{D}_i, \qquad \mathbf{F}_i^{sh,a} = \mathbf{F}_i^{sh} + \mathbf{C}_i - r_i\mathbf{D}_i. \tag{12}$$

Their weighted combination satisfies

$$r_i\mathbf{F}_i^{sh,p} + (1-r_i)\mathbf{F}_i^{sh,a} = \mathbf{F}_i^{sh} + \mathbf{C}_i. \tag{13}$$

Hence, the differential component $\mathbf{D}_i$ cancels, while the shared component $\mathbf{C}_i$ lies in the nullspace of the representative MPP directions. This allows higher-order SH modulation while limiting its response along those directions, without introducing additional trainable parameters.

3.3 Watermark Optimization

We optimize the CPM parameters, including the mixing fractions r_i and residuals $\Delta\mathbf{s}_i$, $\Delta\mathbf{f}_i^{dc}$, and $\mathbf{U}_i$, while keeping the retained base parameters and pretrained watermark decoder $\mathcal{D}$ frozen. For a training view ϕ, let $\mathbf{I}_\phi$ denote the rendering of the original input 3DGS before carrier construction, and $\mathbf{I}_\phi^w$ the corresponding watermarked render. We define

$$\begin{aligned} \mathcal{L}_{\mathrm{vis}} &= (1-\lambda_s)\|\mathbf{I}_\phi^w - \mathbf{I}_\phi\|_1 + \lambda_s\left(1 - \mathrm{SSIM}(\mathbf{I}_\phi^w, \mathbf{I}_\phi)\right), \\ \mathcal{L}_{\mathrm{msg}} &= \mathrm{BCEWithLogits}\left(\mathcal{D}(\mathbf{I}_\phi^w), \mathbf{m}\right), \end{aligned} \tag{14}$$

where $\mathbf{m}$ denotes the target binary message. The overall objective is $\mathcal{L} = \lambda_{\mathrm{vis}}\mathcal{L}_{\mathrm{vis}} + \lambda_{\mathrm{msg}}\mathcal{L}_{\mathrm{msg}}$. The optimization settings and hyperparameters are provided in Appendix C.1.

4 Experiment

We evaluate MPP-GS from four perspectives. We first compare watermarking fidelity and decoding accuracy across three benchmarks and multiple payloads, followed by robustness under common image distortions. We then analyze carrier construction under fixed budgets and finally conduct ablations to isolate the contributions of MPP, CPM, and their components.

4.1 Experimental Setup

Datasets & Metrics. We evaluate the proposed method on three widely used 3DGS benchmarks: Blender (Mildenhall et al., 2021), LLFF (Mildenhall et al., 2019), and Mip-NeRF 360 (Barron et al., 2022). Unless otherwise stated, visual metrics compare renders at matched test poses against the clean input 3DGS. Watermarking results cover all evaluated scenes from the three datasets, with three random seeds per scene. We first average metrics across test views and seeds, then across scenes within each dataset; cross-dataset results are the arithmetic mean of the three dataset-level averages. We evaluate watermarking performance using bit accuracy (BA), PSNR, SSIM, and LPIPS at 32-, 48-, and 64-bit payloads.

Comparison Methods. We compare MPP-GS with representative 3DGS watermarking methods, including GaussianMarker (Huang et al., 2024b), 3D-GSW (Jang et al., 2025), GuardSplat (Chen et al., 2025), WhereWhatWhy (Cai et al., 2026), and W2M (Li et al., 2026). We also construct a baseline to isolate the effectiveness of MPP. Specifically, the baseline uses single-projection packing (SPP), extending UVGS (Rai et al., 2025) to retain complete Gaussian attributes, while keeping all remaining components identical to MPP-GS. Both methods use CPM for watermark embedding and are optimized with the same objectives and optimization settings.

Implementation Details. Unless otherwise stated, we use $K = 10$, with $H = W = 1024$ for Blender and $H = W = 2048$ for LLFF and Mip-NeRF 360. We use $V_{\mathrm{cam}} = 4$ camera-derived directions for Blender and $V_{\mathrm{cam}} = 8$ for LLFF and Mip-NeRF 360, together with the additional directions described in Sec. 3.1. We use the pretrained HiDDeN (Zhu et al., 2018) decoder released by GaussianMarker (Huang et al., 2024b) for watermark decoding. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

Table 1: **Overall watermarking performance averaged across Blender, LLFF, and Mip-NeRF 360.** Baseline denotes SPP+CPM, while MPP-GS denotes MPP+CPM. †W2M uses Scaffold-GS (Lu et al., 2024), whereas all other methods use standard 3DGS (Kerbl et al., 2023). Best and second-best results are shown in **bold** and underline, respectively.

Methods	Publication	32 bits			48 bits			64 bits		
		Bit Acc ↑	PSNR / SSIM ↑	LPIPS ↓	Bit Acc ↑	PSNR / SSIM ↑	LPIPS ↓	Bit Acc ↑	PSNR / SSIM ↑	LPIPS ↓
GaussianMarker	NeurIPS'24	97.89	32.93 / 0.951	0.053	97.78	31.47 / 0.934	0.058	90.84	31.58/ 0.925	0.090
3D-GSW	CVPR'25	97.22	35.15 / 0.977	0.044	93.59	33.26 / 0.972	0.047	91.31	32.52 / 0.966	0.050
GuardSplat	CVPR'25	95.58	35.32 / 0.978	0.043	93.29	33.36 / 0.969	0.045	90.14	32.25 / 0.963	0.048
WhereWhatWhy	CVPR'26	98.46	35.98 / 0.982	0.041	94.29	33.45 / 0.973	0.044	91.65	32.71 / 0.969	0.047
W2M†	CVPR'26	98.34	36.16 / 0.979	0.023	97.80	35.93 / 0.973	0.032	91.17	32.79 / 0.970	0.040
Baseline	–	99.19	34.77 / 0.976	0.027	97.86	32.95 / 0.969	0.046	94.57	30.01 / 0.923	0.051
MPP-GS (Ours)	–	**99.45**	**37.68 / 0.984**	**0.014**	**98.73**	**37.07 / 0.982**	**0.020**	**95.95**	**35.15 / 0.980**	**0.032**

4.2 Main Results

Rendering Fidelity and Bit Accuracy. Table 1 compares watermarking performance at 32-, 48-, and 64-bit payloads. MPP-GS consistently achieves the highest BA, PSNR, and SSIM and the lowest LPIPS across all payloads. Compared with Baseline, its PSNR gain increases from 2.91 dB at 32 bits to 4.12 dB at 48 bits and 5.14 dB at 64 bits, while also achieving higher BA. Since Baseline and MPP-GS use the same CPM embedding and optimization settings and differ only in carrier construction, these gains support the benefit of replacing SPP with MPP. Compared with existing methods, MPP-GS achieves 95.95% BA and 35.15 dB PSNR at 64 bits, versus 91.17% and 32.79 dB for W2M (Li et al., 2026). Figure 3 provides qualitative comparisons at 48 bits across Blender, LLFF, and Mip-NeRF 360. While Baseline and MPP-GS both achieve 100% BA on all three examples, MPP-GS consistently yields higher rendering fidelity, improving PSNR from 32.39 to 36.32 dB on Ship, from 36.89 to 40.00 dB on Flower, and from 34.81 to 37.89 dB on Bonsai. The corresponding residual maps also show visibly smaller distortions, particularly along the hull and base boundaries of Ship, the fine-textured regions of Flower, and the structural edges of Bonsai. These results further illustrate that MPP improves local rendering fidelity under the same CPM embedding scheme.

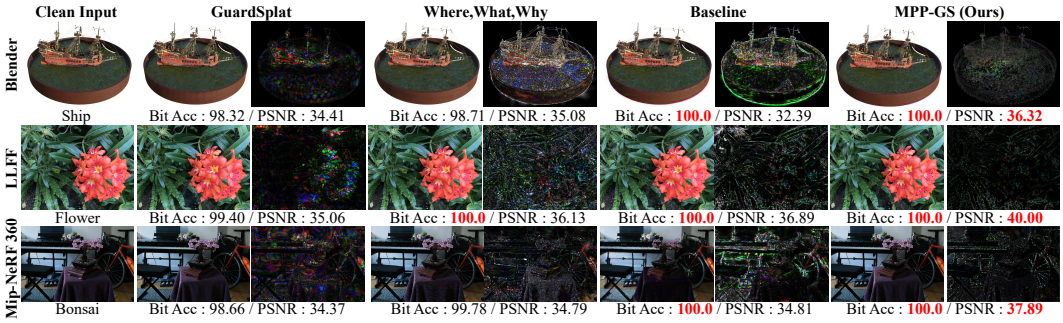
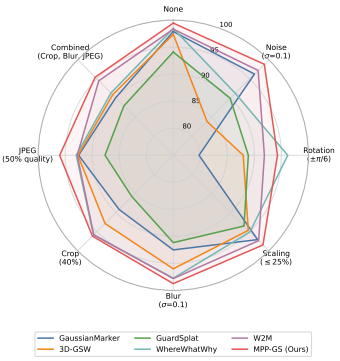
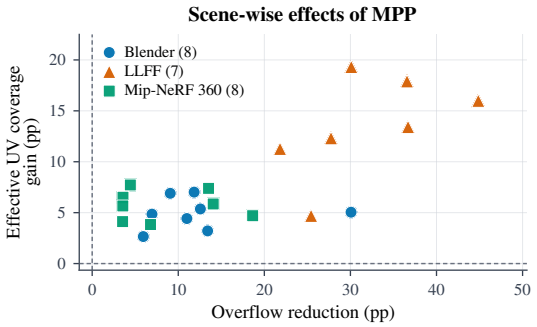


Figure 3: **Qualitative comparison at 48-bit payloads** on Blender, LLFF, and Mip-NeRF 360. Each method is shown with its rendered image and $\times 10$ residual relative to the clean rendering. Although Baseline and MPP-GS both achieve 100% BA, MPP-GS yields smaller residuals and higher PSNR. The improvement is most visible around boundaries, textures, and fine structures, indicating better local rendering fidelity.

Watermark Robustness. Figure 4a evaluates 32-bit watermark extraction under six common 2D distortions and a challenging combined Crop+Blur+JPEG attack. MPP-GS achieves the highest BA in seven of the eight tested conditions and remains second-best under rotation, demonstrating consistently strong robustness across diverse perturbations. In particular, it maintains above 95% BA even under JPEG compression and the combined attack, where several competing methods exhibit noticeably larger degradation. MPP-GS also remains highly stable under noise, scaling, blur, and cropping, indicating that its watermark remains reliably recoverable under both geometric and appearance distortions. Together with the fidelity results in Table 1, these results show that MPP-GS achieves improved rendering quality without sacrificing watermark robustness. Distortion settings and model-level robustness results are provided in Appendices C.4 and G.



(a) Watermark robustness under 2D distortions.



(b) Carrier overflow reduction and coverage gain.

Figure 4: **Watermark robustness and carrier improvements.** (a) Bit accuracy under 2D distortions with 32-bit messages. (b) Changes from SPP to MPP under identical slot budgets. Each point represents one scene, with overflow reduction on the x-axis and effective UV coverage gain on the y-axis, both measured in percentage points. All 23 scenes improve on both metrics.

4.3 Carrier Evaluation

A key goal of MPP is to construct a more balanced and effective carrier before watermark embedding. To isolate carrier construction from CPM, we compare SPP and MPP under identical $H \times W \times K$ budgets before pair allocation. We evaluate primitive retention, spatial distribution, reconstruction fidelity, and construction overhead. We report overflow, defined as the fraction of input primitives excluded during packing, together with clean reconstruction PSNR, core construction time, and peak GPU memory. To characterize the spatial distribution of retained candidates, we define

$$R_{\text{occ}} = \frac{N_{\text{occ}}}{HW}, \qquad R_{\text{eff}} = \frac{\exp\left(-\sum_{p:n_p>0} \omega_p \log \omega_p\right)}{HW}, \qquad \omega_p = \frac{n_p}{N'}, \tag{15}$$

where n_p is the retained occupancy at location p and $N' = \sum_p n_p$. R_{occ} measures occupied UV coverage, while R_{eff} accounts for occupancy imbalance. We also report $R_{\text{sat}}^{\text{prim}}$, the fraction of retained primitives at saturated locations with $n_p = K$. Table 2 shows that MPP consistently reduces overflow, increases UV coverage, and lowers saturation across all three datasets. On LLFF, overflow drops from 32.031% to 0.141%, R_{eff} rises from 4.11% to 17.68%, and $R_{\text{sat}}^{\text{prim}}$ decreases from 47.587% to 4.144%. Figure 4b further shows that all 23 scenes achieve both lower overflow and higher effective coverage. MPP also substantially improves clean reconstruction PSNR, at the cost of higher construction time and memory. Overall, MPP retains more primitives, reduces local saturation, and distributes carrier candidates more broadly across the UV domain.

Table 2: **Carrier construction quality and efficiency under fixed slot budgets.** SPP and MPP use identical $H \times W \times K$ budgets within each dataset. R_{occ} and R_{eff} denote occupied and effective UV coverage, respectively, while $R_{\text{sat}}^{\text{prim}}$ denotes the fraction of retained primitives at saturated locations. All occupancy statistics are measured before pair allocation, and results are averaged over scenes within each dataset.

Dataset	Method	Quality					Efficiency	
		Overflow (%) ↓	R_{occ} (%) ↑	R_{eff} (%) ↑	$R_{\text{sat}}^{\text{prim}}$ (%) ↓	PSNR ↑	Core Time (s) ↓	Peak Mem. (MiB) ↓
Blender	SPP	12.686	10.33	7.44	18.822	50.61	**0.141**	**4472**
	MPP (Ours)	**0.079**	**15.59**	**12.38**	**4.949**	**92.19**	0.317	4512
LLFF	SPP	32.031	5.16	4.11	47.587	35.57	**0.750**	**2972**
	MPP (Ours)	**0.141**	**21.97**	**17.68**	**4.144**	**77.60**	1.539	4372
Mip-NeRF 360	SPP	8.523	27.71	20.31	16.300	52.03	**2.427**	**7392**
	MPP (Ours)	**0.010**	**32.10**	**26.04**	**5.228**	**84.28**	4.573	8692

4.4 Ablation Study

Multi-Projection & Overflow Rescue. Overflow rescue reassigns overflowed Gaussians to alternative UV locations. Table 3(a) evaluates its effect on LLFF under the same carrier budget. Compared with SPP, MPP w/o Rescue reduces overflow from 32.031% to 1.049% and raises PSNR from 35.57 to 61.48 dB. Full MPP further reduces overflow to 0.141% and improves PSNR to 77.60 dB by recovering additional primitives through alternative placements. Core construction time increases from 0.750 to 0.973 s with multi-projection and to 1.539 s with rescue, while rescue adds only 5.9 MiB of peak memory. Overall, multi-projection and rescue improve primitive retention within the fixed carrier capacity, preserving more candidates for subsequent pair allocation.

Table 3: **Component analysis of MPP and CPM.** (a) Effect of multi-projection and overflow rescue on LLFF. (b) Effect of pair parameterization and nullspace modulation on Blender at 48 bits.

(a) Multi-Projection and Overflow Rescue

Variant	Overflow ↓	PSNR ↑	Time (s) ↓	Mem. (MiB) ↓
SPP	32.031%	35.57	0.750	1746.9
MPP w/o Rescue	1.049%	61.48	0.973	2063.6
Full MPP	**0.141%**	**77.60**	1.539	2069.5

(b) Pair Parameterization and Nullspace Modulation

Variant	PP	NM	BA ↑	PSNR ↑	LPIPS ↓
Direct	✗	✗	**98.06**	31.17	0.051
Pair Only	✓	✗	94.95	33.30	0.030
Full CPM	✓	✓	96.19	**34.71**	**0.029**

Pair Parameterization & Nullspace Modulation. Table 3(b) compares Direct, Pair Only, and Full CPM on Blender to evaluate pair parameterization (PP) and nullspace modulation (NM). PP improves PSNR from 31.17 to 33.30 dB and reduces LPIPS from 0.051 to 0.030, although BA decreases from 98.06% to 94.95%. Since higher-order SH updates are not fully canceled by complementary parameterization, NM further constrains their responses along MPP's projection directions. Full CPM raises BA to 96.19% and PSNR to 34.71 dB while further reducing LPIPS to 0.029. These results show that PP improves rendering fidelity, while NM further improves the fidelity-decoding trade-off by reusing MPP's projection directions.

Table 4: **Effect of pair allocation at 48 bits.** Coverage denotes the percentage of feasible UV locations receiving pairs, while Quota CV quantifies pair-allocation imbalance across these locations.

Allocation	Coverage (%) ↑	Quota CV ↓	BA (%) ↑	PSNR ↑	LPIPS ↓
Global Top	73.14	0.8563	**98.89**	35.75	0.03273
Random Quota	73.35	0.8404	98.00	35.57	0.03337
Balanced (Ours)	**100.00**	**0.0648**	98.73	**37.07**	**0.02015**

Pair Allocation. Table 4 compares pair allocation strategies across feasible UV locations. Global Top selects the most important parents globally, Random Quota distributes pairs randomly, and Balanced (Ours) spreads quotas as evenly as possible before selecting important parents within each location. Compared with Global Top, Balanced increases coverage from 73.14% to 100% and reduces quota CV from 0.8563 to 0.0648. It also improves PSNR from 35.75 to 37.07 dB and lowers LPIPS from 0.03273 to 0.02015 at comparable BA (98.89% vs. 98.73%). Random Quota likewise yields lower coverage and rendering fidelity, supporting balanced pair allocation across UV locations.

5 Conclusion

We presented MPP-GS, which organizes pretrained 3DGS into explicit UV units for balanced watermark allocation. Multi-Projection Packing (MPP) provides alternative camera-aware placements to reduce overflow and expand effective UV coverage under a fixed capacity, broadening candidate support for allocation. Complementary Pair Modulation (CPM) exploits the resulting carrier occupancy to allocate pairs and reuses MPP's projection directions to reduce rendering interference. Experiments across three benchmarks and multiple payloads show higher rendering fidelity with comparable or better decoding accuracy. Controlled comparisons with matched-pair budgets further demonstrate the fidelity benefit of balanced location-wise pair allocation. Overall, these results highlight the importance of both constructing a well-distributed carrier and balancing the number of watermark-writing opportunities for high-fidelity 3DGS watermarking.

AI USE STATEMENT

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts and ideas were developed by the authors. Contributions of the LLM focused solely on improving the linguistic quality of the article, without involvement in the scientific content or data analysis.

We take full responsibility for the final content of this work, including the methodology, scientific claims, experimental results, code, and figures. This responsibility includes reviewing AI-assisted content, validating code, and checking reported results against the original experimental records.

ETHICS STATEMENT

This work uses public research benchmarks and does not involve human participants, user studies, or the collection of personal information.

REPRODUCIBILITY STATEMENT

Section 3 describes Multi-Projection Packing, Complementary Pair Modulation, and the watermark optimization procedure. Section 4 specifies the datasets, baselines, implementation settings, and evaluation metrics. Appendices A and B provide additional method details, while Appendix C describes the optimization settings and evaluation protocols.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Conference on Computer Vision and Pattern Recognition*, pp. 5460–5469, 2022.

Mingshu Cai, Jiajun Li, Osamu Yoshie, Yuya Ieiri, and Yixuan Li. Where, what, why: Toward explainable 3d-gs watermarking. In *Conference on Computer Vision and Pattern Recognition*, pp. 20701–20710, 2026.

Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie. Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In *Conference on Computer Vision and Pattern Recognition*, pp. 16325–16335, 2025.

Zijian Dong, Longteng Duan, Jie Song, Michael J. Black, and Andreas Geiger. Moga: 3d generative avatar prior for monocular gaussian avatar reconstruction. In *International Conference on Computer Vision*, pp. 13304–13314, 2025.

Xiufeng Huang, Ka Chun Cheung, Simon See, and Renjie Wan. Geometrysticker: Enabling ownership claim of recolorized neural radiance fields. In *European Conference on Computer Vision*, pp. 438–454, 2024a.

Xiufeng Huang, Ruiqi Li, Yiu-ming Cheung, Ka Chun Cheung, Simon See, and Renjie Wan. Gaussianmarker: Uncertainty-aware copyright protection of 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, pp. 33037–33060, 2024b.

Xiufeng Huang, Ziyuan Luo, Qi Song, Ruofei Wang, and Renjie Wan. Marksplatter: Generalizable watermarking for 3d gaussian splatting model via splatter image structure. In *ACM International Conference on Multimedia*, pp. 12189–12198, 2025.

Youngdong Jang, Dong In Lee, MinHyuk Jang, Jong Wook Kim, Feng Yang, and Sangpil Kim. Waterf: Robust watermarks in radiance fields for protection of copyrights. In *Conference on Computer Vision and Pattern Recognition*, pp. 12087–12097, 2024.

Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting for robust watermarking. In *Conference on Computer Vision and Pattern Recognition*, pp. 5938–5948, 2025.

Xiaoliang Ju and Hongsheng Li. Directtrigs: Triplane-based gaussian splatting field representation for 3d generation. In *Conference on Computer Vision and Pattern Recognition*, pp. 16229–16239, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):1–14, 2023.

Hanyang Kong, Xingyi Yang, and Xinchao Wang. Generative sparse-view gaussian splatting. In *Conference on Computer Vision and Pattern Recognition*, pp. 26745–26755, 2025.

Chenxin Li, Brandon Y Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang. Steganerf: Embedding invisible information within neural radiance fields. In *International Conference on Computer Vision*, pp. 441–453, 2023.

Chenxin Li, Hengyu Liu, Zhiwen Fan, Wuyang Li, Yifan Liu, Panwang Pan, and Yixuan Yuan. Instantsplamp: Fast and generalizable stenography framework for generative gaussian splatting. In *International Conference on Learning Representations*, 2025.

Nan Li, Yike Zeng, Qian Zhang, Qi Zhang, Zhiyi Pan, Wei Feng, and Liang Wan. Write where it matters: Policy-guided watermarks for 3d gaussian splatting. In *Conference on Computer Vision and Pattern Recognition*, pp. 6580–6590, 2026.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-GS: Structured 3D gaussians for view-adaptive rendering. In *Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *International Conference on Computer Vision*, pp. 22401–22411, 2023.

Ziyuan Luo, Anderson Rocha, Boxin Shi, Qing Guo, Haoliang Li, and Renjie Wan. The nerf signature: Codebook-aided watermarking for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4652–4667, 2025.

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):1–14, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Ryutarou Ohbuchi, Akio Mukaiyama, and Shigeo Takahashi. A frequency-domain approach to watermarking 3d shapes. *Computer Graphics Forum*, 21(3):373–382, 2002.

Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1999.

Lingteng Qiu, Shenhao Zhu, Qi Zuo, Xiaodong Gu, Yuan Dong, Junfei Zhang, Chao Xu, Zhe Li, Weihao Yuan, Liefeng Bo, Guanying Chen, and Zilong Dong. Anigs: Animatable gaussian avatar from a single image with inconsistent gaussian reconstruction. In *Conference on Computer Vision and Pattern Recognition*, pp. 21148–21158, 2025.

Aashish Rai, Dilin Wang, Mihir Jain, Nikolaos Sarafianos, Kefan Chen, Srinath Sridhar, and Aayush Prakash. Uvgs: Reimagining unstructured 3d gaussian splatting using uv mapping. In *Conference on Computer Vision and Pattern Recognition*, pp. 5927–5937, 2025.

Aashish Rai, Angela Xing, Anushka Agarwal, Xiaoyan Cong, Zekun Li, Tao Lu, Aayush Prakash, and Srinath Sridhar. Packuv: Packed gaussian uv maps for 4d volumetric video. In *Conference on Computer Vision and Pattern Recognition*, pp. 22581–22593, 2026.

Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. Protecting nerfs' copyright via plug-and-play watermarking base model. In *European Conference on Computer Vision*, pp. 57–73, 2024.

Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Conference on Computer Vision and Pattern Recognition*, pp. 10208–10217, 2024.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Conference on Computer Vision and Pattern Recognition*, pp. 6796–6807, 2024.

Innfarn Yoo, Huiwen Chang, Xiyang Luo, Ondrej Stava, Ce Liu, Peyman Milanfar, and Feng Yang. Deep 3d-to-2d watermarking: Embedding messages in 3d meshes and extracting them from 2d renderings. In *Conference on Computer Vision and Pattern Recognition*, pp. 10021–10030, 2022.

Zhensheng Yuan, Haozhi Huang, Zhen Xiong, Di Wang, and Guanghua Yang. Robust and efficient 3d gaussian splatting for urban scene reconstruction. In *International Conference on Computer Vision*, pp. 26209–26219, 2025.

Youyi Zhan, Tianjia Shao, Yin Yang, and Kun Zhou. Real-time high-fidelity gaussian human avatars with position-based interpolation of spatially distributed mlps. In *Conference on Computer Vision and Pattern Recognition*, pp. 26297–26307, 2025.

Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. In *Advances in Neural Information Processing Systems*, volume 37, pp. 97445–97475, 2024a.

Xuanyu Zhang, Jiarui Meng, Runyi Li, Zhipei Xu, Yongbing Zhang, and Jian Zhang. Gs-hider: Hiding messages into 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, pp. 49780–49805, 2024b.

Haoyu Zhao, Hao Wang, Xingyue Zhao, Hao Fei, Hongqiu Wang, Chengjiang Long, and Hua Zou. Physsplat: Efficient physics simulation for 3d scenes via mllm-guided gaussian splatting. In *International Conference on Computer Vision*, pp. 5242–5252, 2025.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European Conference on Computer Vision*, pp. 682–697, 2018.

Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. In *Advances in Neural Information Processing Systems*, pp. 101790–101817, 2024.

A Additional Details of Multi-Projection Packing

We supplement Sec. 3.1 with the projection construction and packing settings used to form the UV allocation units.

A.1 Multi-Camera Projection

We estimate the scene center $\bar{\mathbf{x}}$ using an opacity-weighted mean of Gaussian positions for Blender and Mip-NeRF 360, and a coordinate-wise median for LLFF. We normalize the camera directions $\mathbf{o}_j - \bar{\mathbf{x}}$ and denote their normalized mean by $\bar{\mathbf{u}}$. Blender and Mip-NeRF 360 use a hybrid set of farthest-point-sampled camera directions and ring directions around $\bar{\mathbf{u}}$. LLFF uses ring directions for the entire camera-derived set.

For orthonormal vectors $\mathbf{e}_1, \mathbf{e}_2$ perpendicular to $\bar{\mathbf{u}}$, a ring of V_{ring} directions at polar angle β is

$$\mathbf{z}_t^{\text{ring}} = \cos\beta\,\bar{\mathbf{u}} + \sin\beta\left(\cos\frac{2\pi t}{V_{\text{ring}}}\,\mathbf{e}_1 + \sin\frac{2\pi t}{V_{\text{ring}}}\,\mathbf{e}_2\right), \quad t = 0, \ldots, V_{\text{ring}} - 1. \tag{16}$$

We add the mean directions $\pm\bar{\mathbf{u}}$ and vertical directions $\pm\mathbf{e}_y$ to obtain the complete set of projection directions $\{\mathbf{z}_v\}_{v=1}^{V}$ in Table 5. All projections share the same carrier.

Table 5: **Projection Direction Configuration.** Total V includes four additional directions, $\pm\bar{\mathbf{u}}$ and $\pm\mathbf{e}_y$.

Dataset	FPS	Ring	β	Total V
Blender	2	2	30°	8
LLFF	0	8	55°	12
Mip-NeRF 360	4	4	30°	12

An orthonormal frame aligned with each projection direction gives the local coordinates in Eq. (2). For $(\tilde{x}, \tilde{y}, \tilde{z})^\top = \tilde{\mathbf{p}}_i^{(v)}$, the spherical projection is

$$\begin{aligned} \theta &= \operatorname{atan2}(\tilde{x}, \tilde{z}), & \varphi &= \arcsin\left(\frac{\tilde{y}}{\|\tilde{\mathbf{p}}_i^{(v)}\|_2}\right), \\ p_x &= \left\lfloor \left(\frac{\theta}{2\pi} + \frac{1}{2}\right)(W-1) \right\rfloor, & p_y &= \left\lfloor \left(\frac{1}{2} - \frac{\varphi}{\pi}\right)(H-1) \right\rfloor, \\ \mathbf{C}[i, v] &= p_y W + p_x. \end{aligned} \tag{17}$$

These projection directions also define the SH constraint in Appendix B.3.

A.2 Importance-Aware Packing

Starting from a randomly selected candidate for each Gaussian, we perform four occupancy-based placement rounds using Eq. (3). Importance-based retention and overflow rescue then follow Sec. 3.1, with replacement margin $\delta = 0$. Each initially overflowed Gaussian is considered once. A displaced Gaussian remains excluded and does not initiate another rescue chain. Pair allocation uses the final occupancy after rescue.

The slot-to-primitive correspondence in Eq. (7) links each retained attribute row to its occupied slot. CPM uses this correspondence to identify the selected parents and associate each auxiliary with a distinct empty slot at the same UV location, as detailed next.

B Additional Details of Complementary Pair Modulation

B.1 Pair Allocation

Let $\mathcal{F} = \{p : c_p > 0\}$ be the feasible UV locations, with c_p from Eq. (8), and let q_p denote the pair count. The budget in Sec. 3.2 satisfies $B_{\text{aux}} \geq |\mathcal{F}|$, since both N_{occ} and $\sum_p c_p$ are at least $|\mathcal{F}|$.

We assign one pair to every feasible location, then increment quotas in rounds over locations with $q_p < c_p$ until $\sum_p q_p = B_{\text{aux}}$. A fixed UV ordering determines which locations receive pairs in a partial final round. Parents are selected by importance within each location, following Sec. 3.2. The allocation remains fixed during watermark optimization.

For example, consider five locations with $K = 10$:

$$\begin{aligned} \text{Occupancy:} \quad & (n_p)_{p=1}^{5} = (10, 9, 4, 1, 0), \\ \text{Pair capacity:} \quad & (c_p)_{p=1}^{5} = (0, 1, 4, 1, 0), \\ \text{Allocated quota:} \quad & (q_p)_{p=1}^{5} = (0, 1, 2, 1, 0). \end{aligned} \tag{18}$$

Here, $N_{\text{occ}} = 4$ and $\sum_p c_p = 6$ give $B_{\text{aux}} = 4$. The three feasible locations receive one pair each, and the third receives the remaining pair. The saturated first location has no free slot, and the empty fifth location has no parent.

Full feasible-location coverage follows from this allocation rule. Its effect on rendering fidelity is evaluated in Table 4. Balanced quotas concern pair counts across UV locations and do not imply equal perturbation magnitudes in rendered images.

B.2 Pair Parameterization

For a parent with opacity α_i, we initialize the pair with equal opacity parameters:

$$\alpha_i^{p,0} = \alpha_i^{a,0} = 1 - \sqrt{1 - \alpha_i}, \qquad r_i^0 = \frac{1}{1 + \sqrt{1 - \alpha_i}}. \tag{19}$$

Substitution into Eq. (9) gives the stated opacity composition. Equal initial opacity parameters therefore do not require $r_i^0 = 1/2$.

We optimize the mixing fraction through a scalar residual,

$$r_i = \sigma\big(\text{logit}(r_i^0) + \eta_i\big), \qquad \eta_i^0 = 0. \tag{20}$$

The DC, higher-order SH, and log-scale residuals also start at zero, so both primitives initially share the original parent's features and scale. These parameter relations do not guarantee unchanged rendered pixels, so fidelity is controlled by the visual loss in Eq. (14).

B.3 Nullspace Modulation

MPP projection directions point outward from the scene center. The renderer uses camera-to-Gaussian SH directions, so we use $\mathbf{d}_v = -\mathbf{z}_v$ to build $\mathbf{Y}$ in Eq. (11). The resulting projector is computed once from the complete set of projection directions, shared across pairs, and fixed during optimization.

The nullspace constraint leaves a shared update space of dimension

$$\dim \ker \mathbf{Y} = L - 1 - \text{rank}(\mathbf{Y}). \tag{21}$$

For degree-three SH, there are $L - 1 = 15$ higher-order coefficients per color channel. The $V = 8$ and $V = 12$ configurations in Table 5 therefore leave at least seven and three shared update dimensions per channel, respectively. These degrees of freedom permit nonzero residuals while suppressing their response along MPP's representative directions.

From Eqs. (11) and (12), the residual response of the weighted pair at direction $\mathbf{d}$ is

$$\Delta \mathbf{c}_i^{sh}(\mathbf{d}) = \mathbf{y}(\mathbf{d})^\top \mathbf{C}_i = \mathbf{y}(\mathbf{d})^\top \mathbf{P}_\perp \mathbf{U}_i. \tag{22}$$

It vanishes at each $\mathbf{d}_v$ because $\mathbf{Y}\mathbf{P}_\perp = \mathbf{0}$, while other directions can retain a nonzero response. This constraint acts on the shared SH response at the chosen directions. Actual camera-to-Gaussian directions and visibility vary across primitives, so the rendering objective remains necessary.

C EXPERIMENTAL DETAILS

Table 6 lists all scenes used in our evaluation. The evaluation and aggregation protocols follow Sec. 4.

Table 6: **Evaluated scenes.** Our experiments cover 23 scenes across three datasets.

Dataset	Count	Scenes
Blender	8	Chair, Drums, Ficus, Hotdog, Lego, Materials, Mic, Ship.
LLFF	7	Fern, Flower, Fortress, Horns, Orchids, Room, Trex.
Mip-NeRF 360	8	Bicycle, Bonsai, Counter, Garden, Kitchen, Room, Stump, Treehill.

C.1 OPTIMIZATION SETTINGS

For the 32-, 48-, and 64-bit MPP-GS experiments, we use Adam with $\epsilon = 10^{-15}$ and set $\lambda_{\text{vis}} = 10$, $\lambda_{\text{msg}} = 1$, and $\lambda_s = 0.2$, where λ_s is the SSIM mixing weight in Eq. (14). The fixed learning rates are 2.5×10^{-3} for the DC residual $\Delta\mathbf{f}_i^{dc}$, 1.25×10^{-4} for the higher-order SH residual $\mathbf{U}_i$, 5×10^{-2} for the mixing residual η_i in Eq. (20), and 5×10^{-3} for the log-scale residual $\Delta\mathbf{s}_i$. Gaussian positions and rotations remain fixed.

We optimize for $4{,}000$, $2{,}000$, and $4{,}000$ iterations on Blender, LLFF, and Mip-NeRF 360, respectively. Each iteration samples one training view. The visual reference is a fixed rendering of the original input PLY before carrier construction, using the same camera and white background as the watermarked rendering. This reference is shared by Baseline and MPP-GS. Decoder preprocessing follows the released GaussianMarker decoder (Huang et al., 2024b). Carrier dimensions and evaluation protocols follow Sec. 4.

C.2 CARRIER EVALUATION

Carrier evaluation uses the final retained carrier after packing and rescue, before pair allocation or watermark optimization. Let N and N' denote the input and retained Gaussian counts, and n_p the occupancy at location p. The overflow and primitive saturation ratios are

$$\text{Overflow} = \frac{N - N'}{N}, \qquad R_{\text{sat}}^{\text{prim}} = \frac{\sum_p n_p \, \mathbf{1}[n_p = K]}{N'}. \tag{23}$$

The latter measures the fraction of retained primitives at full locations, which have no free slot for a local auxiliary. The coverage measures R_{occ} and R_{eff} follow Eq. (15). Both are normalized by HW, and $R_{\text{eff}} \le R_{\text{occ}}$, with equality for equal occupancy among nonempty locations. The four ratios in Table 2 are reported as percentages.

Clean reconstruction PSNR compares the carrier with the original input at matched test poses and backgrounds. Core Time covers the construction core. Peak Mem. in Table 2 additionally includes carrier conversion, unlike the core-only memory measurement in Table 3(a). Dataset averages give each scene equal weight.

For the additional analysis, occupancy evenness J and the potentially pairable fraction η_{pair} are

$$J = \frac{R_{\text{eff}}}{R_{\text{occ}}}, \qquad \eta_{\text{pair}} = \frac{\sum_p \min(n_p, K - n_p)}{N'}. \tag{24}$$

J measures evenness among occupied locations. η_{pair} measures simultaneous pairing capacity under local slot constraints, before applying the auxiliary budget. Both are computed per scene before averaging.

C.3 Pair Allocation Comparison

Table 4 evaluates 48-bit messages on all 23 scenes used in the main paper. Each scene uses three random seeds, matching Table 1. The three allocation strategies share the same MPP carriers, local capacities c_p, total pair budget B_{aux}, CPM parameterization, and optimization settings. Their quotas satisfy $0 \leq q_p \leq c_p$ and $\sum_p q_p = B_{\text{aux}}$. All three select the q_p highest-importance parents within each location.

Global Top. Starting from zero quotas, we traverse retained Gaussians in descending order of ρ_i from Eq. (4). We increment the quota of a Gaussian's location whenever $q_p < c_p$, stopping when the total budget is reached. This implements global importance priority under the local capacity limits.

Random Quota. Pair quotas are sampled from the available local pair capacity under the same total budget and capacity bounds. Feasible locations may receive zero pairs. Randomness changes the location quotas, while parent selection follows the shared importance ranking.

Balanced. We use the capped round-based allocation in Appendix B.1. It first covers every feasible location and distributes the remaining budget across locations with available capacity.

Allocation statistics use the common feasible set $\mathcal{F} = \{p : c_p > 0\}$, determined before allocation. Coverage is the percentage of these locations with $q_p > 0$. Quota CV is the standard deviation of q_p divided by its mean over all of $\mathcal{F}$, including zero-quota locations. Thus, allocation Coverage has denominator $|\mathcal{F}|$, whereas the carrier coverage R_{occ} in Eq. (15) has denominator HW. Statistics are computed for each scene and run before aggregation using the protocol in Sec. 4.

C.4 Robustness Settings

Table 7 summarizes the perturbation levels for our 32-bit robustness evaluations. Image distortions are applied to watermarked renders before decoding. The combined image distortion includes cropping, blurring, and JPEG compression. For model-level robustness, we use the distortion types and levels reported in 3D-GSW (Jang et al., 2025): additive Gaussian noise ($\sigma = 0.1$), removal of 20% of the Gaussians, and cloning of 20%. These distortions are applied separately to the watermarked model before rendering, and BA is evaluated on the resulting images.

Table 7: **Robustness settings.** Image and model distortions correspond to Figure 4a and Table 12, respectively.

(a) 2D image distortions.

Distortion	Level / setting
Noise	$\sigma = 0.1$
Rotation	$\pm\pi/6$
Scaling	$\leq 25\%$
Blur	$\sigma = 0.1$
Crop	40%
JPEG	Quality 50
Combined	Crop, blur, and JPEG

(b) 3D model distortions.

Distortion	Level / setting
Noise	$\sigma = 0.1$
Removing	20%
Cloning	20%

D Carrier Distribution and Pairing Capacity

Table 8: **Additional carrier statistics and reconstruction fidelity.** Statistics are measured before pair allocation and averaged over scenes. J and η_{pair} are defined in Eq. (24). LPIPS compares the clean carrier with the original input.

Dataset	Method	$J \uparrow$	$R_{\text{sat}}^{\text{prim}}$ (%) $\downarrow$	$\eta_{\text{pair}} \uparrow$	LPIPS (10^{-4}) $\downarrow$
Blender	SPP	0.7203	18.82	0.7163	5.657
	MPP	**0.7946**	**4.95**	**0.8936**	**0.0025**
LLFF	SPP	0.7856	47.59	0.3788	131.045
	MPP	**0.8115**	**4.14**	**0.9064**	**0.0734**
Mip-NeRF 360	SPP	0.7398	16.30	0.7266	7.716
	MPP	**0.8179**	**5.23**	**0.8515**	**1.107**

Table 8 complements the main carrier evaluation with evenness and local pairing capacity. MPP increases J on all three datasets, providing more even candidate support. Lower saturation is accompanied by a larger potentially pairable fraction (Figure 5). On LLFF, this fraction rises from 37.88% to 90.64%. These results connect the improved carrier distribution to the local capacity used by CPM. The lower clean-carrier LPIPS also supports better preservation of the input appearance. Potential pairing capacity does not specify the number of pairs ultimately allocated.

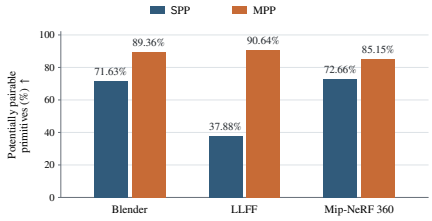


Figure 5: **Potentially pairable primitives.** The fraction of retained primitives that can be paired under local slot constraints, averaged over scenes. Higher values indicate greater theoretical pairing capacity.

E Additional Ablation Study

Additional Projection Directions. Table 9 compares the two additional direction groups with their combination. Full MPP achieves the lowest overflow and highest clean reconstruction PSNR on all three datasets. On Blender, overflow decreases from 0.533% with S1 and 0.398% with S2 to 0.079% with both. The complete configuration therefore provides more effective alternative placements for carrier construction.

Number of Projection Directions. Figure 6 varies V_{cam}. Reconstruction quality generally improves as more projection candidates become available, with smaller gains near the default settings. Blender peaks at four directions, and the LLFF and Mip-NeRF 360 curves largely level off around eight. These results support the selected configurations: increasing the direction count does not consistently improve clean reconstruction.

Table 9: **Additional projection directions.** S1 adds $\pm\bar{\mathbf{u}}$, S2 adds $\pm\mathbf{e}_y$, and Full MPP adds both to the camera-derived directions.

Dataset	Setting	PSNR↑	Overflow (%)↓
Blender	S1	86.60	0.533
	S2	88.03	0.398
	Full MPP	**92.19**	**0.079**
LLFF	S1	73.32	0.201
	S2	73.96	0.200
	Full MPP	**77.60**	**0.141**
Mip-NeRF 360	S1	81.62	0.037
	S2	81.55	0.029
	Full MPP	**84.28**	**0.010**

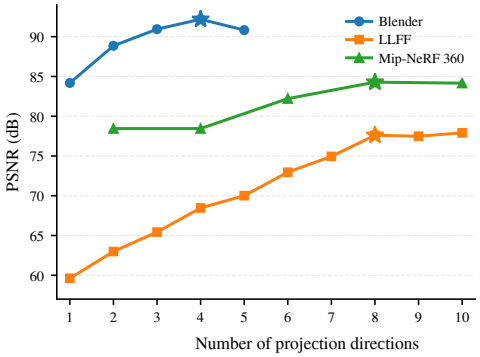


Figure 6: **Effect of the number of camera-derived projection directions.** Stars mark the default configurations.

F Efficiency and Model Overhead

F.1 Carrier Preparation and Embedding Time

Table 10 reports carrier preparation separately from watermark optimization. The reported 8.70 s preparation time is short relative to the 8 min embedding stage. Embedding is faster than the reported times for 3D-GSW, GuardSplat, and W2M, and slightly slower than GaussianMarker. This comparison characterizes the processing cost of introducing the structured carrier alongside the fidelity results.

Table 10: **Carrier preparation and embedding time.**

Method	Carrier (s)↓	Embedding (min)↓
GaussianMarker	–	7
3D-GSW	–	12
GuardSplat	–	15
W2M	–	9
MPP-GS	8.70	8

F.2 Model Size and Rendering Overhead

Table 11 quantifies the storage and rendering cost of auxiliary Gaussians. MPP-GS uses larger watermarked models and higher rendering-time ratios than GaussianMarker on Blender and LLFF, but smaller models and a lower ratio on Mip-NeRF 360. Its rendering time is 1.481–1.816 times that of the corresponding clean model. These costs accompany the fidelity gains and vary with the asset and the allocated auxiliary count.

Table 11: **Model size and rendering overhead.** We report average auxiliary counts N_{aux} in thousands, auxiliary-to-clean count ratios R_{aux}, watermarked PLY sizes, and WM/Clean rendering-time ratios. GM denotes GaussianMarker.

Dataset	Method	N_{aux}	R_{aux} (%)	PLY (MiB)	$t_{\mathrm{WM}}/t_{\mathrm{Clean}}$
Blender	GM	122.44	42.69	96.96	$1.344\times$
	MPP-GS	163.47	58.32	106.32	$1.816\times$
LLFF	GM	760.52	47.93	601.98	$1.535\times$
	MPP-GS	921.53	56.96	639.57	$1.729\times$
Mip-NeRF 360	GM	1,739.19	48.85	1,284.83	$1.592\times$
	MPP-GS	1,339.71	45.23	1,190.05	$1.481\times$

G Model Distortion Robustness

Table 12 extends the image-distortion evaluation to model-level noise, removal, and cloning at a 32-bit payload. MPP-GS achieves 90.01%, 97.52%, and 98.21% BA, respectively. It ranks first under cloning and second under noise and removal among the reported methods. These results complement the high-fidelity watermarking evaluation with recoverability under the tested model perturbations.

Table 12: **Model distortion robustness.** BA (%) at 32 bits.

Method	Clean	Noise $\sigma = 0.1$	Removing 20%	Cloning 20%
GaussianMarker	97.89	87.12	97.48	97.45
3D-GSW	97.22	89.93	97.23	96.85
GuardSplat	95.58	79.96	88.49	92.30
WhereWhatWhy	<u>98.46</u>	**90.52**	**98.16**	<u>97.48</u>
MPP-GS	**99.45**	<u>90.01</u>	<u>97.52</u>	**98.21**

H Additional Qualitative Results

Figures 7–9 show carrier construction and 48-bit watermark embedding for all evaluated scenes. The two residual columns show the visual changes after MPP and after CPM at the same $\times 10$ display scale.

Blender. Figure 7 shows weak carrier-construction residuals across the eight objects. Watermark residuals remain visible around fine structures and textured surfaces, including Drums, Lego, and Ship. The rendered objects retain their overall appearance, while the residuals make the remaining changes visible.

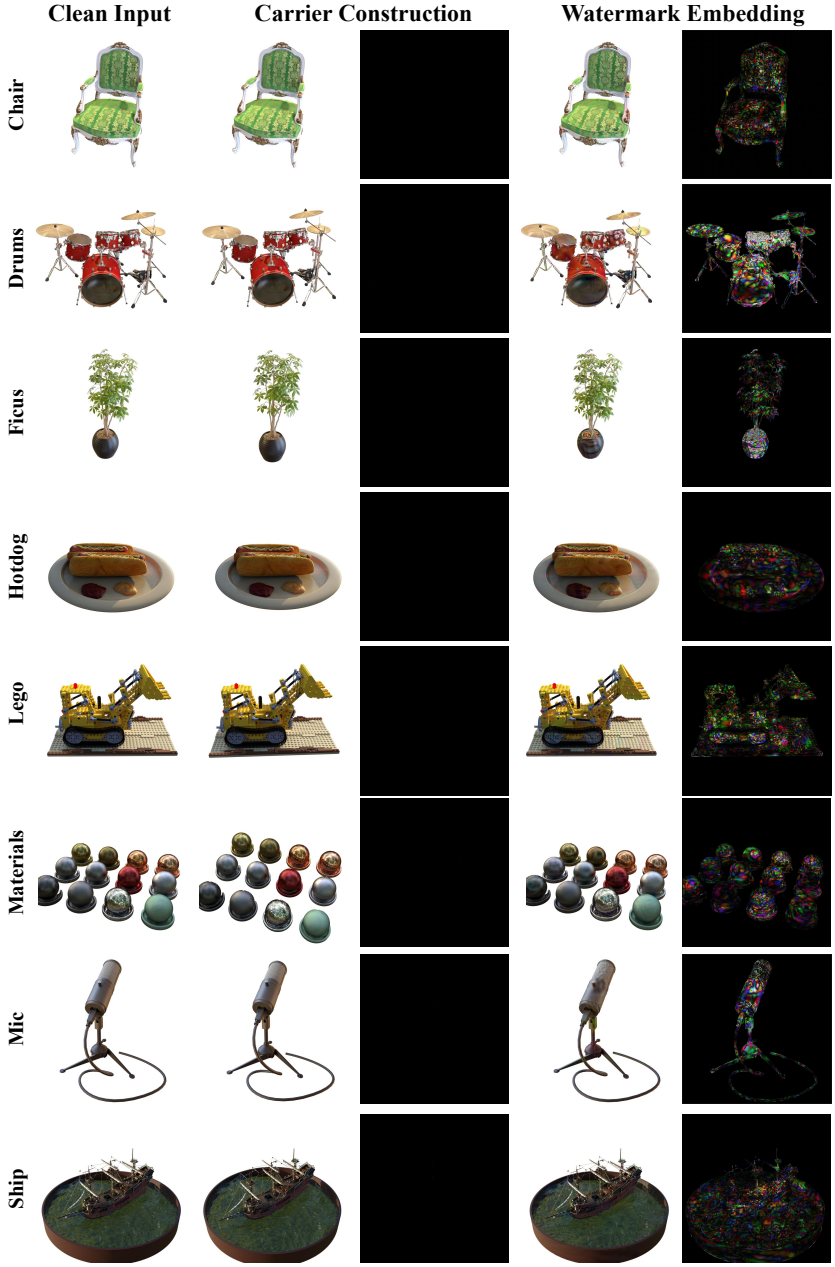


Figure 7: **Carrier construction and 48-bit watermark embedding on Blender.** All eight evaluated scenes are shown. From left to right: the clean input rendering, the rendering after MPP and its $\times 10$ residual map, and the watermarked rendering after CPM and its $\times 10$ residual map.

LLFF. Figure 8 extends the visualization to seven forward-facing scenes. Carrier-construction residuals are weak in the displayed views. After watermark embedding, residuals are more apparent along plant details in Fern and Flower and structural boundaries in Room and Trex, illustrating the remaining appearance changes across different scene contents.

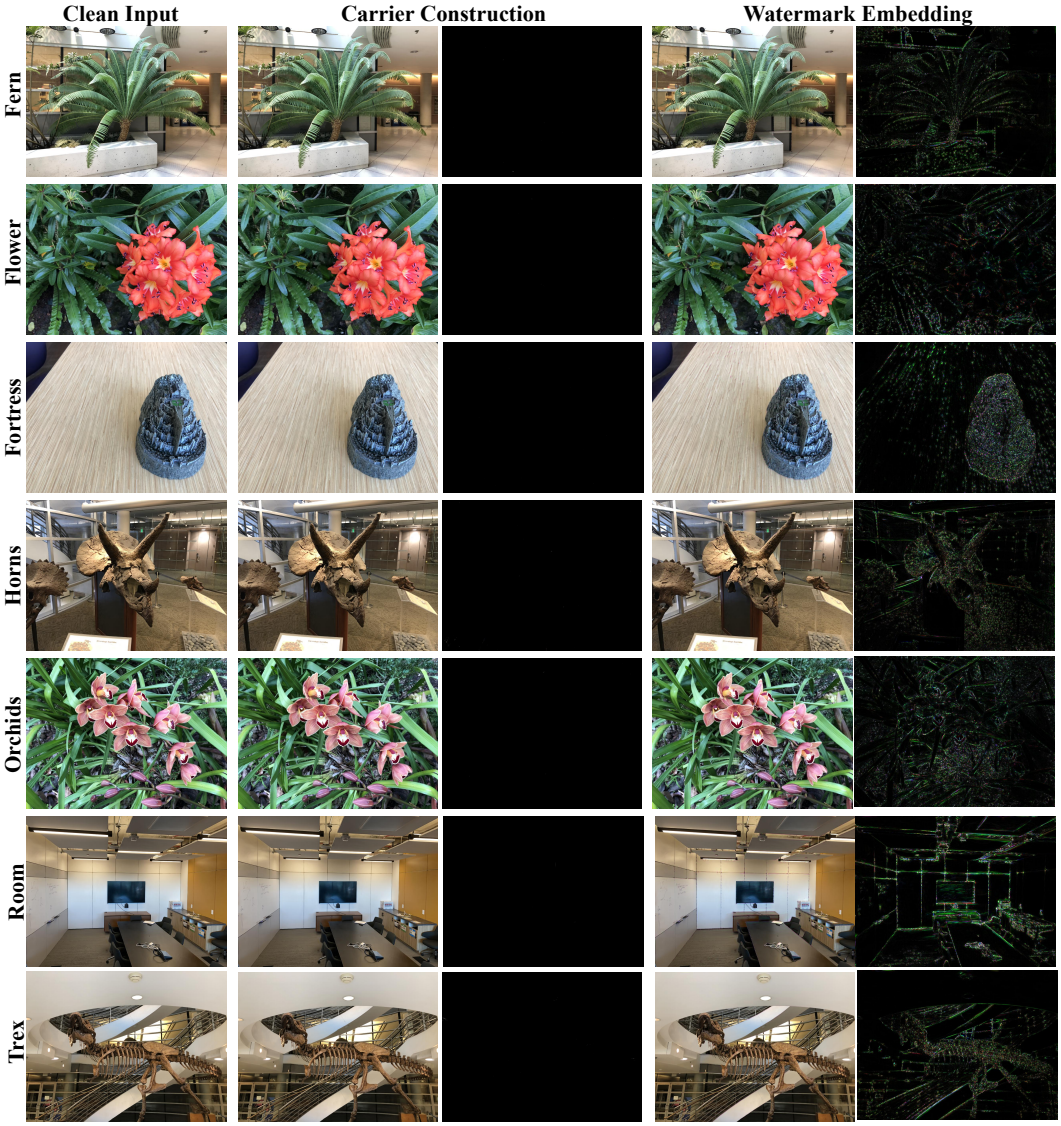


Figure 8: **Carrier construction and 48-bit watermark embedding on LLFF.** All seven evaluated scenes are shown, with the same column layout as Figure 7.

Mip-NeRF 360. Figure 9 covers eight indoor and outdoor scenes. Carrier-construction residuals are visible in examples such as Bicycle and Stump, showing that MPP does not eliminate reconstruction error. Watermark residuals also vary across fine geometry and textured regions. These examples complement the aggregate fidelity metrics with the scene-dependent visual effects of the two stages.



Figure 9: **Carrier construction and 48-bit watermark embedding on Mip-NeRF 360.** All eight evaluated scenes are shown, with the same column layout as Figure 7.