

GuardSplat: Efficient and Robust Watermarking for 3D Gaussian Splatting

Supplementary Material

A. Overview

In this supplementary material, we further provide more discussions, implementation details, and results as follows:

- Section B depicts the architecture of our message decoder guided by CLIP [41], and we also conduct a comparison for watermarking efficiency against the state-of-the-art methods.
- Section C conducts an additional evaluation for security, exploring whether the watermarks can be simply removed from model files.
- Section D illustrates the visualization results of various ablations in Tables 3 and 4 of the main paper.
- Section E reports more results, including the quantitative results on larger-capacity messages $N_L = \{64, 72\}$, bit accuracy across various rendering situations, and the zoomed-in rendering results between watermarked and original views.

B. Decoder Architecture and Watermarking Speed

As shown in Fig. S1, our message decoder only consists of 3 fully-connected (FC) layers, which can accurately map the CLIP textual features to the corresponding binary messages after a 5-minute optimization. Thanks to CLIP’s rich representation, our decoder can achieve excellent performance with minimal parameter size. We also investigate the watermarking efficiency between our *GuardSplat* and state-of-the-art methods. As shown in the training accuracy curve in Figure S2, our *GuardSplat* achieves the highest efficiency, which only takes 10 minutes to watermark a pre-trained 3DGS asset.

C. Additional Evaluation for Security

We conduct additional experiments to evaluate the security of our *GuardSplat* in Table S1, investigating whether the malicious users can remove the watermarks from the model file by pruning the $K\%$ of Gaussians, where $K \in \{5, 10, 15, 20, 25\}$. “Bottom K ” denotes pruning K of low-opacity Gaussians, while “random” denotes randomly pruning K of the Gaussians. As demonstrated, our *GuardSplat* still achieves a bit accuracy of 98.74% when 25% of the low-opacity Gaussians are removed, indicating that simply removing low-opacity Gaussians does not effectively attack our method. Though randomly removing the Gaussians can lead to a significant decline in bit accuracy, it also greatly affects the reconstruction quality (*i.e.*, PSNR, SSIM, and LPIPS), resulting in low-fidelity rendering. This experi-

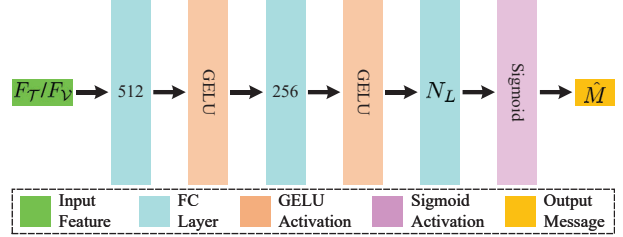


Figure S1. **The architecture of our message decoder.** Given an output feature F_T or F_V , we first pass it through two FC layers with GELU activations, where their channels are set to 512 and 256, respectively. Then, we map the feature to the binary message using a N_L -channel FC layer and a Sigmoid activation.

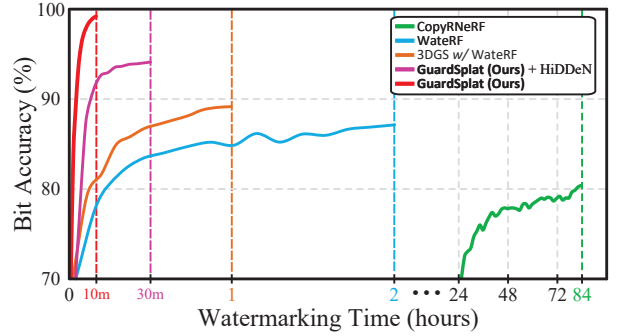


Figure S2. **Training accuracy curves** with $N_L = 32$ bits on Blender [32] dataset. Our *GuardSplat* achieves high training efficiency, which only takes 10 minutes to watermark a 3D asset.

mental result demonstrates that the malicious cannot directly remove the watermarks from the model file, verifying the security of our *GuardSplat*.

D. Additional Visual Comparisons

D.1. Various Message Embedding Strategies

In the main paper, we explore the performance under various message embedding strategies with $N_L = 32$ bits (see quantitative results in Table 3). For better comparisons, we further visualize the results of various message embedding strategies in Figure S3. As shown, the proposed SH-aware module achieves superior bit accuracy and reconstruction quality to the competitors.

D.2. Various Loss Combinations

In the main paper, we quantitatively compare the performance across various loss combinations in Table 4. We also conduct a visual comparison of these ablation variants in

Table S1. **Security analysis across various pruning ratios $K\%$.** Bottom K denotes removing $K\%$ of the low-opacity Gaussians, while Random denotes randomly removing $K\%$ of the Gaussians.

$\%$	Bottom K				Random			
	Bit Acc	PSNR	SSIM	LPIPS	Bit Acc	PSNR	SSIM	LPIPS
5	99.04	39.38	0.9939	0.0022	98.59	37.76	0.9916	0.0033
10	99.02	39.06	0.9937	0.0025	96.87	36.35	0.9891	0.0047
15	98.99	38.68	0.9933	0.0031	94.68	35.14	0.9832	0.0063
20	98.94	38.33	0.9928	0.0037	91.98	33.98	0.9779	0.0081
25	98.74	37.87	0.9922	0.0041	88.59	31.50	0.9721	0.0103

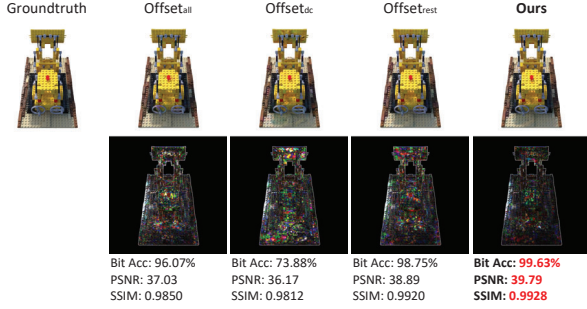


Figure S3. **Visual comparisons between various message embedding strategies and our SH-aware module.** Heatmaps at the bottom show the differences ($\times 10$) between the watermarked and Groundtruth. **Red** text indicates the best performance.

Figure S4. As shown, “ $\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{msg}} + \mathcal{L}_{\text{off}}$ ” achieves the best performance in bit accuracy and reconstruction quality.

E. More Results

E.1. Quantitative Results on Larger-Capacity Messages

To further investigate the superiority of our *GuardSplat* in capacity, we supplement the results on larger message lengths ($N_L \in \{64, 72\}$) in Table S2. As demonstrated, the bit accuracy and reconstruction quality of our 72-bit results are still higher than the state-of-the-art methods on $N_L \in \{16, 32, 48\}$ bits reported in the main paper (see Table 1), significantly improve the capacity of existing baselines.

E.2. Bit Accuracy across Various Rendering Situations

We explore the extraction accuracy of learned SH offsets across the following situations: **1) SH Noise**; **2) Light Conditions**; **3) Occlusions**; and **4) Viewing Angles**. Specifically, to simulate different lighting conditions, we first train a 3DGS asset of “Lego” from the TensorIR [16] dataset in “RGBA” mode. We then freeze all Gaussian attributes while optimizing the SH features to adapt to various illumination scenarios, such as “light”, “sunset”, and “city”.

Table S2. **Quantitative results of our *GuardSplat* on Blender [32] and LLFF [31] datasets with $N_L \in \{64, 72\}$ bits.**

N_L	Bit Acc	PSNR	SSIM	LPIPS
64	97.41	37.76	0.9899	0.0040
72	96.64	36.47	0.9866	0.0053



Figure S4. **Visual comparisons of various loss combinations.** “Ours” denotes the combination of $\mathcal{L}_{\text{msg}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{off}}$. Heatmaps at the bottom show the differences ($\times 10$) between the watermarked and Groundtruth. **Bold** text indicates the best overall performance.

We train only the SH offsets in “RGBA” mode and add them to the SH features of other lighting modes for evaluation. As shown in Figure S5, *GuardSplat* achieves good robustness against SH noise (a) and light conditions (b) by adding noise to SH features in training. Since the occluded areas can be removed by segmentation models (e.g., Segment Anything Model [21], and Grounding DINO [25]), we train *GuardSplat* to extract messages from randomly masked views ($\leq 20\%$). It improves the robustness of our *GuardSplat* against various occlusions (c). *GuardSplat* is inherently robust to various viewing angles (d) since it is designed for 3D.

E.3. Zoomed-in Rendering Results

Since SH features produce highly realistic shading and shadowing, altering them may reduce fidelity, especially in the specular areas. To clearly show how the SH offsets are changing the rendering results, we conduct a visual comparison of zoomed-in rendering results between the original 3DGS and our *GuardSplat* of “ball” on the Shiny [54] dataset in Figure S6. As shown, *GuardSplat* can preserve the original metallic luster of assets.

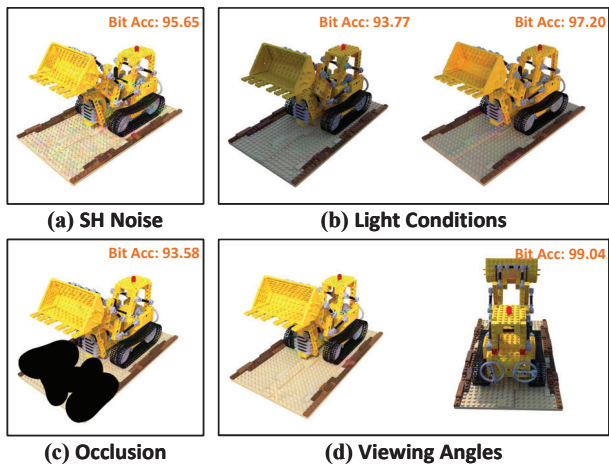


Figure S5. Bit accuracy across various rendering parameters.



Figure S6. Zoomed-in rendering results between the original 3DGS and our *GuardSplat*.