This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

GuardSplat: Efficient and Robust Watermarking for 3D Gaussian Splatting

Zixuan Chen[1] Guangcong Wang[2] Jiahao Zhu[1] Jianhuang Lai[1,3,4,5] Xiaohua Xie[1,3,4,5*]
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]School of Computing and Information Technology, Great Bay University, Dongguan, China
[3]Guangdong Province Key Laboratory of Information Security Technology, China
[4]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[5]Pazhou Lab (Huangpu), Guangzhou, China
{chenzx3, zhujh59}@mail2.sysu.edu.cn, wanggc3@gmail.com, {stsljh, xiexiaoh6}@mail.sysu.edu.cn

Project Page: https://narcissusex.github.io/GuardSplat Code: https://github.com/NarcissusEx/GuardSplat

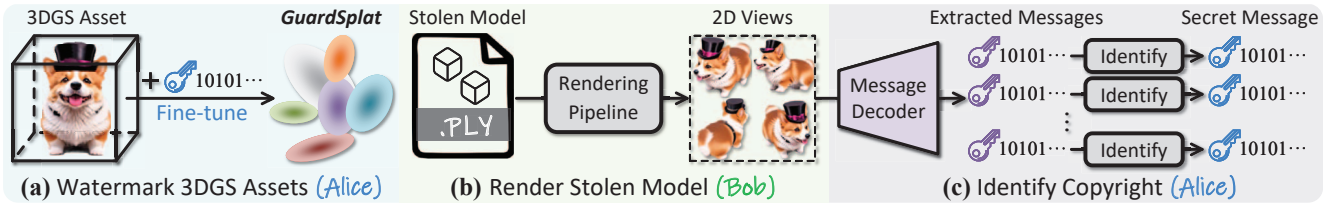

Figure 1. **Application scenarios of GuardSplat.** To protect the copyright of 3D Gaussian Splatting (3DGS) [19] assets, **(a)** the owners (Alice) can use our **GuardSplat** to embed the secret message (blue key) into these models. **(b)** If malicious users (Bob) render views for unauthorized uses, **(c)** Alice can use the private message decoder to extract messages (purple key) for copyright identification.

Abstract

3D Gaussian Splatting (3DGS) has recently created impressive 3D assets for various applications. However, considering ***security***, ***capacity***, ***invisibility***, *and training* ***efficiency***, *the copyright of 3DGS assets is not well protected as existing watermarking methods are unsuited for its rendering pipeline. In this paper, we propose* ***GuardSplat***, *an innovative and efficient framework for watermarking 3DGS assets. Specifically,* ***1)*** *We propose a CLIP-guided pipeline for optimizing the message decoder with minimal costs. The key objective is to achieve high-accuracy extraction by leveraging CLIP's aligning capability and rich representations, demonstrating exceptional* ***capacity*** *and* ***efficiency****. 2) We tailor a Spherical-Harmonic-aware (SH-aware) Message Embedding module for 3DGS, seamlessly embedding messages into the SH features of each 3D Gaussian while preserving the original 3D structure. This enables watermarking 3DGS assets with minimal fidelity trade-offs and prevents malicious users from removing the watermarks from the model files, meeting the demands for* ***invisibility*** *and* ***security****. 3) We present an Anti-distortion Message Extraction module to improve* ***robustness*** *against various distortions. Experiments demonstrate that* ***GuardSplat*** *outperforms state-of-the-art and achieves fast optimization speed.*

*Corresponding author.

1. Introduction

3D representation is a cutting-edge technique in computer vision and graphics, playing a vital role in various domains such as film production, game development, virtual reality, and autonomous driving. One of the most promising approaches in this field is 3D Gaussian Splatting (3DGS) [19]. 3DGS revolutionizes 3D representation techniques by offering high fidelity, rapid optimization capabilities, and real-time rendering speed, which enables the creation of impressive 3D assets [11, 24, 51, 59, 62, 63] in the real world. However, the risk of valuable 3DGS assets being stolen by unauthorized users poses significant losses to creators. This situation raises an urgent question: *How can we design a method tailored for 3DGS to protect copyright?*

One effective strategy for copyright protection is embedding secret messages into 3DGS assets. However, it has several challenging requirements: **1) Security:** A secure watermark should be difficult to detect and cannot be removed from the model. **2) Invisibility:** Any watermarked view rendered from a 3D asset should visually maintain consistency with the corresponding view rendered from original models, avoiding disruption of normal use. **3) Capacity:** Large-capacity messages can be effectively embedded into a 3DGS model and accurately extracted from 2D rendered views. **4) Efficiency:** Fast optimization speed is essential to meet real-world demands. When considering all of these

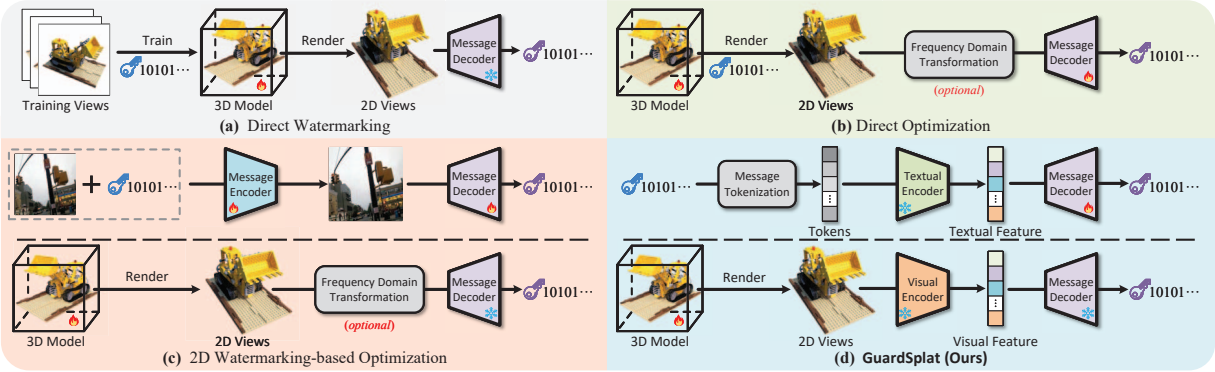

Figure 2. **Comparisons of four 3D watermarking frameworks**. They differ in how to embed messages and train message decoders. **(a)** Directly training 3D models on the watermarked images. **(b)** Simultaneously training a 3D model and a message decoder. **(c)** Employing the message decoder from a 2D watermarker for optimization. **(d) GuardSplat** first trains a message decoder to extract messages from CLIP [41] textual features. This message decoder then can be applied to the CLIP visual features for watermarking 3D models via optimization.

four important requirements, it is challenging to design a good 3D watermarking method for 3DGS.

Existing watermarking methods [14, 28, 66, 69] partially meet the four requirements above and have improved a lot for the watermark of 2D or 3D digital assets. However, they are inadequate for the 3DGS framework, which can be categorized into three groups. **First**, an intuitive method is directly applying 2D watermarking methods for 2D training or rendered views. For instance, employing HiDDeN [69] to watermark the 2D rendered views cannot protect the original model files. Besides, directly training 3D models on the watermarked images shown as Figure 2 **(a)** exhibits low bit accuracy in message extraction, as it cannot guarantee novel views contain a consistent watermark. **Second**, some methods directly embed messages into 3D models during optimization (*e.g.,* [28, 66], Figure 2 **(b)**). Although it guarantees a consistent watermark across novel views, it optimizes a new message decoder per scene during watermarking 3D models, requiring expensive optimization costs. **Third**, to avoid per-scene optimization, one might consider a general-purpose message decoder (*e.g.,* [13–15, 49], Figure 2 **(c)**), which is pre-trained from a 2D watermarking network. However, these networks are encoder-decoder models that simultaneously reconstruct the image and extract the message, where the encoder tries to keep the fidelity between the input and output images while the decoder subsequently tries to extract the messages as intact as possible from the output image. As a result, directly using the decoder to watermark 3D models may yield degraded performance due to the fidelity-capacity trade-off. Moreover, simultaneously optimizing both encoder and decoder is time-consuming.

In this paper, we present *GuardSplat*, a novel watermarking framework to protect the copyright of 3DGS assets. As shown in Figure 2 **(d)**, compared to conventional 2D watermarking methods that optimize both the message encoder and decoder simultaneously, we propose a message decoupling optimization module guided by Contrastive Language-Image Pre-training (CLIP) [41], which can only train the decoder for message extraction (top row), significantly reducing the optimization costs. Thanks to the text-image aligning capability and rich representations of CLIP, this decoder can embed large-capacity messages into 3DGS assets (bottom row) with minimal costs (see Figures 3 and S2), demonstrating superior **capacity** and **efficiency**. Subsequently, we tailor a message embedding module for 3DGS, employing a set of spherical harmonics (SH) offsets to seamlessly embed messages into the SH features of each 3D Gaussian while maintaining the original 3D structure. This enables watermarking 3DGS assets with minimal trade-offs in fidelity and also prevents malicious users from removing the watermarks from the model files, meeting the demands for **invisibility** and **security**. We further design an anti-distortion message extraction module, which simulates the randomly distorted views during optimization using the differentiable distortion layer, allowing the watermarked SH features to achieve strong **robustness** against various distortions. Extensive experiments on Blender [32] and LLFF [31] datasets demonstrate that our *GuardSplat* outperforms the state-of-the-art methods. *GuardSplat* achieves fast optimization speed, which takes 5 and 10 minutes to train the decoder and watermark a 3DGS asset on a single RTX 3090 GPU, respectively.

Overall, the main contributions of this paper are summarized as follows: **1)** We present *GuardSplat*, a new watermarking framework to protect the copyright of 3DGS assets. **2)** We propose a CLIP-guided Message Decoupling Optimization module to train the message decoder, achieving superior **capacity** and **efficiency**. We tailor a message embedding method for 3DGS to meet the **invisibility** and **security** demands. We further introduce an anti-distortion message extraction for good **robustness**. **3)** Experiments demonstrate that our *GuardSplat* outperforms state-of-the-art and achieves fast optimization speed for training message decoder and watermarking 3DGS assets.

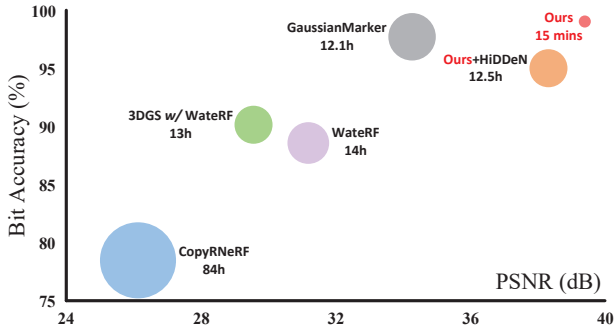

Figure 3. **Performance of state-of-the-art methods** with $N_L = 32$ bits on Blender [32] and LLFF [31] datasets. The radius of circles is proportional to their total training time (decoder optimization + watermarking) evaluated on RTX 3090 GPU.

2. Related Works

3D Representations. Neural radiance field (NeRF) [32] is a compelling solution for 3D representations, which is based on the standard volumetric rendering [18] and alpha compositing techniques [39], building the implicit representations using the multi-layer perceptron (MLP). Follow-up works adapt NeRF to various domains, such as sparse-view reconstruction [8, 55, 61], acceleration [33], generative modeling [35, 46], text-to-3D generation [38, 57], anti-aliasing [4, 5], medical image super-resolution [7], and RGB-D scene synthesis [2]. 3D Gaussian Splatting (3DGS) [19] has become a mainstream approach for 3D representations with its fast optimization and rendering speed. Unlike NeRF, 3DGS explicitly represents the scene using a set of 3D Gaussians and renders views through splatting [22]. It has been applied to various scenarios, including text-to-3D generation [24], avatar generation [47, 63], single-view generation [51, 71], anti-aliasing [62], and SLAM [30, 59].

Digital Watermarking. Early studies [3, 23, 34, 42, 53] proposed to embed the watermarks within frequency domains. With the advent of deep learning, Zhu *et al*. [69] proposed the first end-to-end deep watermarking framework – HiDDeN, while the subsequent advances investigate to improve the robustness [1, 9, 26, 29], and extend the application scenarios [27, 52, 64]. Recent methods [10, 58] proposed diffusion-based watermarking to protect the contents yielded from diffusion models [12, 44, 45, 48]. To protect 3D assets, most methods [36, 40, 67, 70] focused on embedding and detecting watermarks within meshes and point clouds. Yoo *et al*. [60] provided a novel perspective, which embedded the invisible watermarks into 3D models through differentiable rendering pipelines, allowing the watermarks to be extracted from rendered views. Inspired by [60], CopyRNeRF [28], WateRF [14], and NeRFProtector [49] aim to insert watermarks into NeRF [32]. Specifically, CopyRNeRF [28] replaces the color representations, while WateRF [14] and NeRFProtector [49] embed the message into model weights via optimization. Recently, Song *et al*. [50] propose to prevent using Triplane Gaussian Splatting (TGS) [71] for unauthorized 3D reconstruction from copyrighted images. Zhang *et al*. [66] introduced a steganography model for 3DGS that employs secured features to replace SH features, and trains scene and message decoders to extract views and hidden messages, respectively. Current works [13, 15] aim to directly embed messages into 3DGS models via a pre-trained 2D watermarking decoder. However, since 2D watermarking methods have an inherent trade-off between fidelity and capacity, using their decoder for optimization may result in sub-optimal capacity. Moreover, [13, 15] may significantly alter the 3D structure during watermarking, leading to low-fidelity results.

3. Preliminary

3D Gaussian Splatting. 3DGS [19] is a recent groundbreaking method for novel view synthesis. Given the center position $\boldsymbol{\mu} \in \mathbb{R}^3$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^7$, a 3D Gaussian at position $\mathbf{x}$ can be queried as follows:

$$\mathcal{G}(\mathbf{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})). \quad (1)$$

Then, given the projective transformation $\mathbf{P}$, viewing transformation $\mathbf{W}$, and Jacobian $\mathbf{J}$ of the affine approximation of $\mathbf{P}$, the corresponding 2D mean position $\hat{\boldsymbol{\mu}}$ and covariance $\hat{\boldsymbol{\Sigma}}$ of the projected 3D Gaussian can be calculated as:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{PW}\boldsymbol{\mu}, \quad \hat{\boldsymbol{\Sigma}} = \mathbf{JW}\boldsymbol{\Sigma}\mathbf{W}^\top\mathbf{J}^\top. \quad (2)$$

Let $\hat{C} \in \mathbb{R}^{W\times H\times 3}$ denote a $W \times H$ RGB view rendered by 3DGS, the color of each pixel (x, y) can be generated as:

$$\hat{C}_{x,y} = \sum_{i=1}^{N} \boldsymbol{c}_i \sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \quad \sigma_i = \alpha_i \mathcal{G}((x,y) : \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}), \quad (3)$$

where N represents the number of Gaussians overlaping the pixel (x, y). $\boldsymbol{c}_i \in \mathbb{R}^3$ and $\alpha_i \in \mathbb{R}^1$ denote the color transformed from k-ordered spherical harmonic (SH) coefficients $\boldsymbol{h}_i \in \mathbb{R}^{3\times(k+1)^2}$ and opacity of the i-th Gaussian, respectively. Since the rendering pipeline is differentiable, 3DGS models can be optimized by the loss function as:

$$\mathcal{L}_{\text{rgb}} = \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}}(\hat{C}, C) + (1-\lambda_{\text{ssim}})\mathcal{L}_1(\hat{C}, C), \quad (4)$$

where C is a groundtruth image and λ_{ssim} is set to 0.2.

Contrastive Language-Image Pre-training. CLIP [41] is pre-trained to match images with natural language descriptions on 400 million training image-text pairs collected from the internet. It consists of two independent encoders: a textual encoder $\mathcal{E}_\mathcal{T}$ and a visual encoder $\mathcal{E}_\mathcal{V}$, which extract textual features $F_\mathcal{T} \in \mathbb{R}^{512}$ and visual features $F_\mathcal{V} \in \mathbb{R}^{512}$ from the given batch of images and texts, respectively. These encoders are trained to learn the aligning capability of text-image pairs by maximizing the similarity between textual and visual features via a contrastive loss.

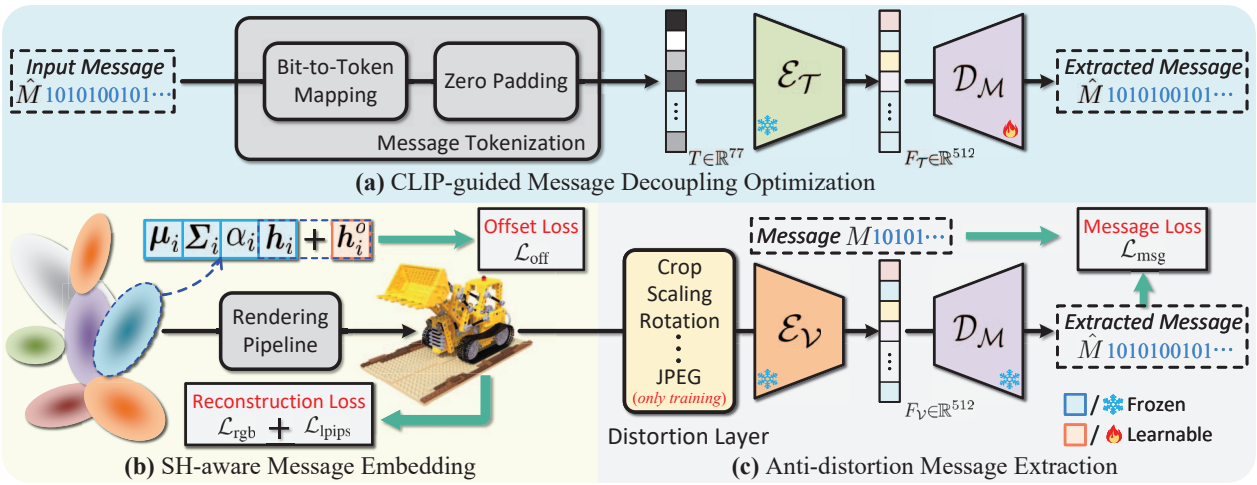

Figure 4. **Overview of GuardSplat. (a)** Given a binary message $M \in \{0,1\}_{i=1}^{L}$, we first transform it into CLIP tokens T using the proposed message tokenization. We then employ CLIP's textual encoder $\mathcal{E}_\mathcal{T}$ to map T to the textual feature $F_\mathcal{T}$. Finally, we feed $F_\mathcal{T}$ into message decoder $\mathcal{D}_\mathcal{M}$ to extract the message $\hat{M} \in \{0,1\}_{i=1}^{L}$ for optimization. **(b)** For each 3D Gaussian, we freeze all the attributes and build a learnable spherical harmonic (SH) offset $\boldsymbol{h}_i^o$ as the watermarked SH feature, which can be added to the original SH features as $\boldsymbol{h}_i + \boldsymbol{h}_i^o$ to render the watermarked views. **(c)** We first feed the 2D rendered views to CLIP's visual encoder $\mathcal{E}_\mathcal{V}$ to acquire the visual feature $F_\mathcal{V}$ and then employ the pre-trained message decoder to extract the message $\hat{M}$. A differentiable distortion layer is used to simulate various visual distortions during optimization. $\mathcal{D}_\mathcal{M}$ and $\boldsymbol{h}_i^o$ are optimized by Eq. (7) and Eq. (10), respectively.

4. Method

In this section, we propose *GuardSplat* to effectively protect the copyright of 3D Gaussian Splatting (3DGS) [19] assets. The overview of the proposed method is depicted in Figure 4. Specifically, we first propose a message decoupling optimization module guided by Contrastive Language-Image Pretraining (CLIP) [41] to train the message decoder $\mathcal{D}_\mathcal{M}$. By analyzing the 3DGS rendering pipeline, we then present a spherical-harmonic-aware (SH-aware) message embedding module to integrate the watermarked SH features for a pre-trained 3DGS model. Furthermore, we design a strategy for anti-distortion message extraction. As a result, *GuardSplat* can not only embed large-capacity messages with minimal optimization costs but also achieves superior fidelity and robustness, suggesting that *GuardSplat* can protect the copyright of 3DGS models without the affection of normal use.

4.1. Message Decoupling Optimization

As discussed in recent works [14, 28, 60], one of the representative approaches to watermark 3D assets is embedding the messages into a 3D representation model via optimization. However, these methods encounter limitations in efficiency. Specifically, one group of methods aims to directly embed messages into 3D models for optimization, as shown in Figure 2 **(b)**. They optimize a message decoder per scene, which is time-consuming. To learn a general-purpose message decoder, the other group of methods first trains an encoder and decoder such that it can reconstruct the image and extract the message given an image and message as input. The message decoder is then used for 3D watermarking as shown in Figure 2 **(c)**. Since conventional methods optimize both the message encoder and decoder simultaneously, it also takes much time for optimization.

To address this issue, we propose a Message Decoupling Optimization module guided by CLIP [41] that optimizes a general-purpose message decoder and a 3DGS model, as shown in Figure 2 **(d)**. The key to the success of this module is that the Contrastive Language-Image Pre-training (CLIP) [41] builds a bridge between the texts and images. As discussed in Section 3, CLIP consists of a good textual encoder and a good visual encoder that is trained on a dataset of 400 million text-image pairs, providing rich text and image representations. As shown in Figure 4 **(a)**, given a binary message $M \in \{0,1\}_{i=1}^{L}$, we first transform it into CLIP tokens T using the proposed bit-to-token mapping as:

$$T = \{t_S\} \cup \left\{ \bigcup_{i=1}^{L} \Phi(M_i, i) \right\} \cup \{t_E\}, \tag{5}$$

where $t_S = 49406$ and $t_E = 49407$ denote the start and ending points of the CLIP text token, respectively. $\Phi(\cdot,\cdot)$ is a function that uniformly maps the i-th bit to an integer number within the range $[1, 49405]$. To match the format of CLIP tokens, T is then zero-padded to a size of 77. Finally, we feed the tokens into CLIP textual encoder $\mathcal{E}_\mathcal{T}$ and employ a message decoder $\mathcal{D}_\mathcal{M}$ built by multi-layer perception (MLP) with 3 fully-connected (FC) layers to extract the messages $\hat{M}$ from the output textual features $F_\mathcal{T} \in \mathbb{R}^{512}$ as:

$$\hat{M} = \mathcal{D}_\mathcal{M}(\mathcal{E}_\mathcal{V}(T)). \tag{6}$$

$\mathcal{D}_{\mathcal{M}}$ can be optimized by minimizing the message loss as:

$$\mathcal{L}_{\text{msg}} = -\sum_{i=1}^{L} M_i \log \hat{M}_i + (1 - M_i) \log(1 - \hat{M}_i). \quad (7)$$

As a result, the message decoder can be directly optimized without being constrained by invisibility, achieving superior capacity and training efficiency. Furthermore, CLIP's rich representation also enables it to achieve better performance with minimal optimization costs.

4.2. SH-aware Message Embedding

Unlike neural radiance fields (NeRF) [32], 3D Gaussian Splatting (3DGS) [19] explicitly represents the scene through a set of 3D Gaussian as Eq. (1) and renders the views using Eq. (3). Given the i-th 3D Gaussian, it consists of 4 attributes: center position $\boldsymbol{\mu}_i$, covariance $\boldsymbol{\Sigma}_i$, opacity $\boldsymbol{\alpha}_i$, and spherical harmonic (SH) feature $\boldsymbol{h}_i$, where the former three attributes denote the 3D structure and while the latter is related to the color representation. An intuitive solution to watermark 3DGS assets is directly updating all the attributes of 3D Gaussians during optimization. However, it may significantly alter the 3D structure (i.e., $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, $\boldsymbol{\alpha}_i$), which leads to sub-optimal fidelity while concealing large-capacity messages (see "Offset$_{\text{all}}$" in Table 3).

Based on the above observations, we argue that it is required to maintain the original 3D structure during optimization. To achieve this, we propose an SH-aware Message Embedding module, a simple yet efficient approach tailored for 3DGS to watermark pre-trained models with minimal losses in fidelity. As shown in Figure 4 **(b)**, for each 3D Gaussian, we freeze all the attributes and create a learnable SH offset $\boldsymbol{h}_i^o \in \mathbb{R}^{48}$ for watermarking. The reason behind this is the fact that SH parameters represent view-dependent effects like glossy or specular highlights, which only exist in a few regions of a scene. Thus, embedding the secret message into SH features with minimal constraints can preserve the fidelity of the 3D asset. Specifically, we first add each SH offset $\boldsymbol{h}_i^o$ to the corresponding SH coefficient $\boldsymbol{h}_i$ as the watermarked SH feature, and then feed it into the 3DGS rasterization to render the watermarked views. To further alleviate the fidelity decline brought by excessive offset, we employ an offset loss to constrain its magnitude:

$$\mathcal{L}_{\text{off}} = -\frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{h}_i^o\|_2^2, \quad (8)$$

where N denotes the number of 3D Gaussians. As a result, the secret message can be seamlessly embedded into the SH offset of each 3D Gaussian in optimization, which not only maintains the original 3D structure but also prevents malicious users from removing the watermarks from the model files, achieving superior invisibility and security.

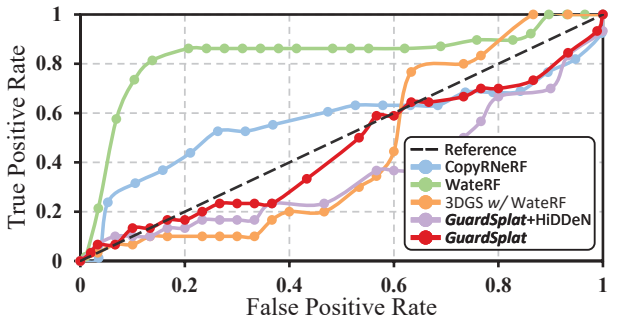

Figure 5. **ROC curves** produced by varying thresholds in StegExpose [6] on different methods. The closer the curve is to the "Reference", the more effective the method is regarding security.

4.3. Anti-distortion Message Extraction

Given CLIP's visual encoder $\mathcal{E}_{\mathcal{V}}$ and the message decoder $\mathcal{D}_{\mathcal{M}}$ pre-trained in Section 4.1, we can directly extract the hidden messages $\hat{M}$ from the watermarked views $\hat{C}$ based on CLIP's aligning capability by:

$$\hat{M} = \mathcal{D}_{\mathcal{M}}(\mathcal{E}_{\mathcal{V}}(\hat{C})), \quad (9)$$

Thanks to the generalization capability of CLIP, the watermarked color features have strong robustness against the visual distortions of Gaussian blur, and Gaussian noise. However, it struggles to deal with other visual distortions such as rotation and JPEG compression. To robustly extract messages from the watermarked views across various types of distortions, we further propose an anti-distortion message extraction module. In Figure 4 **(c)**, we employ a differentiable distortion layer to randomly simulate some visual distortions on the rendered views during optimization. These distortions include cropping, scaling, rotation, JPEG compression, and brightness jittering, enabling the watermarked SH features to learn the anti-distortion ability against most visual distortions via optimization. The distortion layer is only used during training. After optimization, *GuardSplat* can deal with most visual distortions, achieving superior extraction accuracy under challenging conditions.

4.4. Full Objective

For optimization, we first freeze CLIP's visual and textual encoders and train the message decoder by minimizing the message loss $\mathcal{L}_{\text{msg}}$ in Eq. (7) between the input and extracted messages. After pre-training the CLIP-guided message decoder, we then employ it to watermark the pre-trained 3DGS models. We freeze the message decoder and utilize it to extract the message from the rendered views, and the secret message can be embedded into 3DGS models by minimizing the following loss:

$$\mathcal{L} = \lambda_{\text{recon}}(\mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{lpips}}) + \lambda_{\text{msg}}\mathcal{L}_{\text{msg}} + \lambda_{\text{off}}\mathcal{L}_{\text{off}}, \quad (10)$$

where λ_{recon}, λ_{msg}, and λ_{off} are the hyper-parameters to balance the corresponding terms. Since $\mathcal{L}_{\text{rgb}}$ and $\mathcal{L}_{\text{lpips}}$ sepa-

Table 1. **Comparisons of the start-of-the-art methods** on Blender [32] and LLFF [31] datasets for bit accuracy and reconstruction qualities *w.r.t* various message lengths. **Bold** text indicates the best performance in this table.

Methods	16 bits				32 bits				48 bits			
	Bit Acc	PSNR	SSIM	LPIPS	Bit Acc	PSNR	SSIM	LPIPS	Bit Acc	PSNR	SSIM	LPIPS
NeRF-based Watermarking Methods												
CopyRNeRF [28]	91.16	26.29	0.9100	0.0380	78.08	26.13	0.8960	0.0410	60.06	27.56	0.8950	0.0660
WateRF [14]	95.67	32.79	0.9480	0.0330	88.58	31.19	0.9360	0.0400	85.82	30.86	0.9300	0.0400
3DGS Built on Watermarked Images												
3DGS [19] + CIN [29]	56.73	31.95	0.9194	0.1027	53.13	31.74	0.9279	0.0944	55.78	30.25	0.9139	0.1120
3DGS [19] + SSL [9]	58.94	36.51	0.9737	0.0094	61.85	35.24	0.9706	0.0179	58.79	35.88	0.9710	0.0123
3DGS [19] + HiDDeN [69]	63.07	31.59	0.9790	0.0171	52.46	34.51	0.9682	0.0209	53.08	33.42	0.9687	0.0299
3DGS [19] + DwtDctSvd [34]	55.44	34.78	0.9582	0.0399	53.15	32.32	0.9477	0.0547	51.83	31.09	0.9302	0.0704
3DGS [19] + StegaStamp [52]	79.72	35.52	0.9697	0.0181	82.36	34.04	0.9601	0.0265	83.97	32.54	0.9523	0.0406
3DGS optimized by altering all attributes ($\text{Offset}_{\text{all}}$)												
GaussianMarker [13]	99.36	34.42	0.9822	0.0124	98.85	33.98	0.9788	0.0163	98.25	32.12	0.9723	0.0234
3DGS [19] *w/* WateRF [14]	92.89	31.01	0.9678	0.0475	90.15	29.56	0.9611	0.0492	87.30	29.13	0.9562	0.0534
GuardSplat (Ours) + 2D Watermarking Decoders												
GuardSplat (Ours) + CIN [29]	95.75	37.88	0.9762	0.0092	93.35	37.42	0.9726	0.0109	92.77	37.10	0.9689	0.0124
GuardSplat (Ours) + SSL [9]	99.50	40.92	0.9935	0.0020	98.60	38.95	0.9920	0.0028	98.14	38.51	0.9909	0.0030
GuardSplat (Ours) + HiDDeN [69]	98.75	40.48	0.9909	0.0025	95.58	38.32	0.9897	0.0025	93.29	38.56	0.9886	0.0032
GuardSplat (Ours) + StegaStamp [52]	99.00	38.55	0.9903	0.0035	98.28	38.63	0.9914	0.0030	97.23	38.27	0.9892	0.0037
GuardSplat (Ours)	**99.64**	**41.55**	**0.9957**	**0.0017**	**99.04**	**39.40**	**0.9939**	**0.0022**	**98.29**	**38.90**	**0.9923**	**0.0028**

rately denote the RGB loss in Eq. (4) and LPIPS loss [17], optimizing the reconstruction term $\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{lpips}}$ can improve the visual similarity between the watermarked and original views. For extraction, given a 3DGS model watermarked by our *GuardSplat*, we can render the views from arbitrary viewpoints using the official rendering pipeline, while the hidden messages can be directly extracted from the rendered views by Eq. (9) for copyright identification.

5. Experiments

Datasets. Following the settings in [14, 28], we choose two commonly-used datasets: the Blender [32] and LLFF [31] datasets, for evaluation. Specifically, the Blender dataset consists of 8 synthetic bounded scenes, while the LLFF dataset consists of handheld forward-facing captures of 8 real scenes. For each scene, we employ 200 test views to evaluate the visual quality and bit accuracy. We report average values across all testing views in our experiments.

Baselines. We compare *GuardSplat* against six baselines to ensure a fair comparison: **1)** CopyRNeRF [28], **2)** WateRF [14], **3)** GaussianMarker [13], **4)** 3DGS *w/* WateRF, **5)** 3DGS trained on watermarked images (*i.e.,* 3DGS + 2D watermarking methods), and **6)** *GuardSplat* optimized by other pre-trained 2D watermarking decoders, (*i.e., GuardSplat* + 2D watermarking decoders). Specifically, CopyRNeRF and WateRF are NeRF-based [32] watermarking methods, while GaussianMarker is a 3DGS watermarking approach. 3DGS *w/* WateRF refers to applying WateRF to 3DGS models. The 2D watermarking methods include DwtDctSvd [34], StegaStamp [52], SSL [9], and CIN [29].

Implementation Details. *GuardSplat* is implemented on the top of a Pytorch [37] implementation of 3DGS [19][1]. Our experiments run on a single RTX 3090 GPU. We first train a 3DGS model on a given scene with multi-view photos and then embed the message into that pre-trained 3DGS model. The size of learnable SH offsets is equal to that of the SH coefficients in the pre-trained models.

For training the message decoder, we employ Adam [20] as the optimizer with a weight decay of 10^{-6} and a batch size of 64. The maximum epoch is set to 100, and the learning rate is set to 5×10^{-3} as default. Given the message length N_L, we randomly select $\texttt{min}(2^{N_L}, N_K)$ samples from a total of 2^{N_L} messages as training and test data, where N_K=2048. It only takes 5 minutes for optimization. Thanks to CLIP's rich representation, our decoder achieves excellent performance with only 3 FC layers, detailed architecture is provided in Supp. B.

For watermarking 3DGS models, we employ Adam [20] as the optimizer with a weight decay of 10^{-6} and a batch size of 16. The maximum epoch is set to 100, and the learning rate of the SH offsets is set to 5×10^{-3}. The hyperparameters in Eq. (10) are set as λ_{recon}=1, λ_{msg}=0.03, and λ_{off}=10, respectively. It takes 10 minutes for watermarking.

For Distortion Layer, since the proposed CLIP-guided message decoder can deal with Gaussian Noise and Gaussian Blur, we only consider the visual distortions of cropping, scaling, rotation, brightness jittering, JPEG compression, and VAE attacks. The former four visual distortions are differentiably re-implemented by Pytorch [37] built-in functions, while the differentiable JPEG comparison is built by [43]. During training, we employ the differentiable distortion layer to simulate various visual distortions for opti-

[1] https://github.com/graphdeco-inria/gaussian-splatting

Table 2. **Comparisons of the start-of-the-art methods** on Blender [32] and LLFF [31] datasets for bit accuracy *w.r.t* various distortion types. We show the results on 16-bit messages. **Bold** text indicates the best performance in this table.

Methods	None	Noise (μ=0.1)	Rotation ($\pm\pi/6$)	Scaling ($\leq$25%)	Blur (σ=0.1)	Crop (40%)	Brightness (0.5$\sim$1.5)	JPEG (10% quality)	VAE Attack [68] Bmshj2018	Combined
CopyRNeRF [28]	91.16	90.04	88.13	89.33	90.06	–	–	–	–	–
WateRF [14]	95.67	95.36	93.13	93.29	95.25	95.40	90.91	86.99	51.73	84.12
3DGS [19] *w/* WateRF [14]	92.89	87.35	88.28	90.33	91.92	89.07	88.71	88.49	55.48	86.37
GaussianMarker [13]	99.36	99.13	70.84	97.89	94.40	98.52	95.78	86.22	52.00	83.49
GuardSplat (Ours) + CIN [29]	95.75	94.87	90.89	94.50	95.16	93.82	93.97	88.61	49.25	84.03
GuardSplat (Ours) + SSL [9]	99.50	99.57	86.78	84.53	98.79	77.54	94.31	92.99	47.42	74.85
GuardSplat (Ours) + HiDDeN [69]	98.75	96.63	90.02	95.93	94.87	97.25	94.97	90.04	53.14	88.70
GuardSplat (Ours) + StegaStamp [52]	99.00	98.38	53.21	95.17	98.17	51.34	95.48	88.81	80.12	64.75
GuardSplat (Ours)	**99.64**	**99.60**	**94.56**	**98.75**	**99.27**	**98.71**	**97.46**	**94.70**	**82.35**	**93.38**

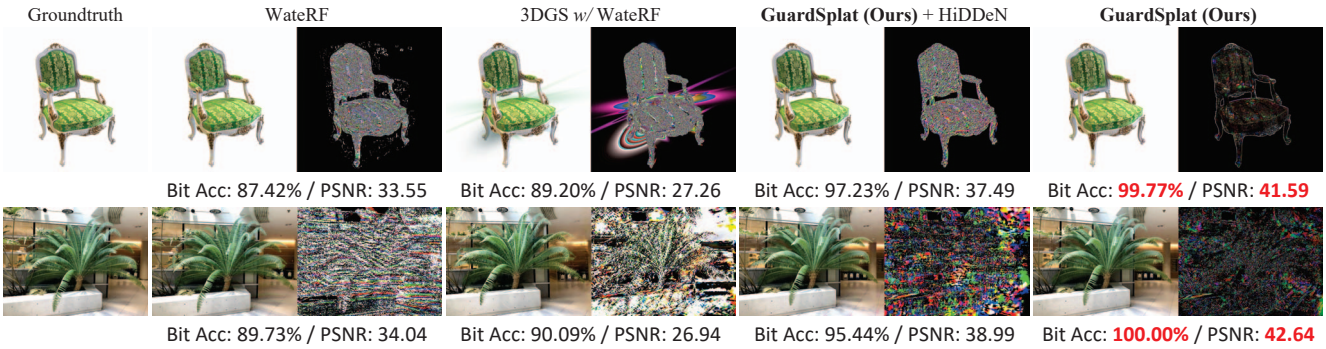

Figure 6. **Visual comparisons** with $N_L = 32$ bits. Heatmaps show the differences ($\times 10$) between the watermarked and original views.

mization. In the test, the rendered views are distorted using the visual distortions built by OpenCV for evaluation.

Evaluation Metrics. We follow the standard of digital watermarking to evaluate *GuardSplat* and the baselines in five aspects: **1) Capacity:** We evaluate the bit accuracy across various message lengths $N_L \in \{16, 32, 48\}$ on the 2D rendered views. **2) Invisibility:** We evaluate the visual similarity of views rendered from the watermarked and original models using Peak Signal-to-Noise Ratio (PSNR), Structured Similarity Index (SSIM) [56], and Learned Perceptual Image Patch Similarity (LPIPS) [65]. **3) Robustness:** We investigate the extraction accuracy with $N_L = 16$ bits across various visual distortions, including Gaussian Noise of μ=0 and σ=0.1, Random Rotation of angles within $[-\frac{\pi}{6}, +\frac{\pi}{6}]$, Random Scaling of ratios within $[0.75, 1.25]$, Gaussian Blur of *kernel_size*=3 and σ=0.1, 40% Center Crop, Brightness Jittoring of ratios within $[0.5, 1.5]$, JPEG compression of 10% image quality, VAE attack [68] based on Bmshj2018, and the combination of Crop, Brightness, and JPEG. **4) Security:** We perform StegExpose [6], an LSB steganography detection, on the rendered views. The detection set is built by mixing the watermarked and original views with equal proportions. **5) Efficiency:** We analyze the relationship between bit accuracy and training time.

5.1. Experimental Results

Security. We claim that our *GuardSplat* is secure. Since it adaptively embeds messages by slightly perturbing the SH features of every 3D Gaussian, it is difficult to remove the watermark from the model file. To further verify the security, we employ StegExpose [6] to detect steganographic content in rendered views. Figure 5 depicts the detection results on our *GuardSplat* and the baselines. As shown, our *GuardSplat* achieves superior security to the competitors.

Capacity & Invisibility. We report the capacity and invisibility across various message lengths $N_L \in \{16, 32, 48\}$ on the Blender [32] and LLFF [31] datasets in terms of bit accuracy, PSNR, SSIM, and LPIPS in Table 1. As demonstrated, our *GuardSplat* surpasses all the competitors with a consistently superior performance *w.r.t* various message lengths, achieving a significant improvement in extraction accuracy and invisibility. Moreover, *GuardSplat* achieves higher bit accuracy than *GuardSplat* + 2D watermarking decoders with $N_L \geq 32$ bits, indicating that our CLIP-guided message decoder is better at handling large-capacity messages. Besides, *GuardSplat* + HiDDeN outperforms 3DGS *w/* WateRF as they share the same decoder, proving the superiority of our SH-aware Message embedding module.

Robustness. We report bit accuracy across various visual distortions with $N_L = 16$ bits in Table 2. Visually, our *GuardSplat* outperforms all the baselines across various distortions, demonstrating that our Anti-distortion Message Extraction module enables the watermarked 3DGS models to learn robust SH features during optimization.

Efficiency. As shown in Figure 3, the efficiency of existing advances is unsatisfactory. Specifically, training a CopyRNeRF requires 84 hours, while WateRF, and 3DGS *w/* WateRF separately spend 14 and 13 hours for optimization as

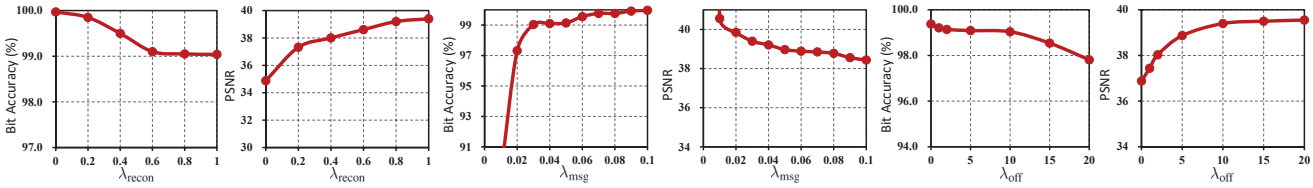

Figure 7. **Performance across different hyper-parameter values** with $N_L = 32$ bits on the Blender [32] and LLFF [31] datasets.

Table 3. **Various embedding methods** on Blender [32] and LLFF [31] datasets with N_L=32 bits. **Bold** text denotes the best score.

	Bit Acc	PSNR	SSIM	LPIPS
$\text{Offest}_{\text{all}}$	98.79	36.56	0.9804	0.0123
$\text{Offset}_{\text{dc}}$	74.59	36.98	0.9828	0.0146
$\text{Offest}_{\text{rest}}$	98.25	38.70	0.9892	0.0077
SH-aware (Ours)	99.04	39.40	0.9939	0.0022

it takes 12 hours to train the HiDDeN. Compared to these methods, our *GuardSplat* achieves much higher efficiency, which only takes 5 and 10 minutes to train the message decoder and watermark a 3DGS asset, respectively. We also provide the comparison of watermarking speed in Supp. B.

Visual Comparison. We visually compare our *GuardSplat* against baselines with $N_L = 32$ bits in Figure 6. As shown, our results present superior reconstruction quality and bit accuracy to the competitors. Moreover, the fidelity of 3DGS *w/* WateRF is much lower than *GuardSplat* + HiDDeN, which demonstrates that altering all the attributes during optimization may lead to a significant decline in image quality.

5.2. Ablation Study & Sensitivity Analysis

We first conduct ablation experiments to prove the effectiveness of our model designs, including various message embedding strategies and loss combinations. Subsequently, we evaluate the bit accuracy and fidelity of our method across different values of λ_{recon}, λ_{msg}, and λ_{off} to analyze the sensitivity. The results are evaluated on the Blender [32] and LLFF [31] datasets with $N_L = 32$ bits.

Various message embedding Strategies. We compare our SH-aware message embedding module with three strategies in Table 3, including updating all the attributes: "$\text{Offest}_{\text{all}}$", the DC components of SH features: "$\text{Offset}_{\text{dc}}$", and the residuals of SH features: "$\text{Offest}_{\text{rest}}$". As shown, $\text{Offest}_{\text{all}}$ achieves unsatisfactory reconstruction quality (row 1) as it alters the original 3D structure to embed messages during optimization. Moreover, $\text{Offset}_{\text{dc}}$ (row 2) and $\text{Offest}_{\text{rest}}$ (row 3) are inferior to our SH-aware module (row 4), proving the correctness of our motivation.

Different Loss Combinations. We explore the optimal loss combinations in Table 4. Compared to the original 3DGS model (row 1), only using message loss $\mathcal{L}_{\text{msg}}$ (row 2) will significantly reduce the reconstruction quality. Though simultaneously minimizing the message loss $\mathcal{L}_{\text{msg}}$ and reconstruction loss $\mathcal{L}_{\text{recon}}$ (row 3) can alleviate the decline in fidelity, it can only achieve sub-optimal results due to some large SH offsets. Thus, we design the offset loss $\mathcal{L}_{\text{off}}$ to restrict the deviation of SH offsets, which achieves the optimal reconstruction quality (row 4).

Table 4. **Different loss combinations** with N_L=32 bits on Blender [32] and LLFF [31] datasets. The first row denotes the original 3DGS, and $\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{rgb}} + \mathcal{L}_{\text{lpips}}$ indicates the reconstruction loss.

$\mathcal{L}_{\text{msg}}$	$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{\text{off}}$	Bit Acc	PSNR	SSIM	LPIPS
			53.41	*inf*	1.0000	0.0000
✔			100.00	31.79	0.9604	0.0379
✔	✔		99.26	36.88	0.9831	0.0101
✔	✔	✔	99.04	39.40	0.9939	0.0022

Various Hyper-parameter Values. We analyze the sensitivity of 3 hyper-parameters: λ_{recon}, λ_{msg}, and λ_{off} in Figure 7. For simplification, we only change the value of one hyper-parameter, while keeping the other two at their default values. Visually, as increasing λ_{recon} from 0 to 1, the PSNR rises with a subtle decline in bit accuracy. When $\lambda_{\text{msg}} \in [0, 0.03]$, there is a significant change in performance. However, this effect gradually diminishes as $\lambda_{\text{msg}} \in [0.03, 0.1]$. $\lambda_{\text{off}} = 10$ is a watershed in performance, as values above and below it will influence the trade-off. Thus, we choose $\lambda_{\text{recon}} = 1$, $\lambda_{\text{msg}} = 0.03$, and $\lambda_{\text{off}} = 10$ for the optimal overall performance.

6. Conclusion

In this paper, we present *GuardSplat*, a novel watermarking framework to protect the copyright of 3DGS assets. Specifically, we build an efficient message decoder via CLIP-guided Message Decoupling Optimization, enabling high-capacity and efficient 3DGS watermarking. Moreover, we tailor a SH-aware message embedding module for 3DGS to seamlessly embed the messages while maintaining fidelity, meeting the demands for invisibility and security. We further propose an anti-distortion message extraction module to achieve strong robustness against various visual distortions. Experiments demonstrate that our *GuardSplat* outperforms the baselines and achieves fast training speed.

Acknowledgement. This project is partially supported by the NSFC (U22A2095), the Major Key Project of PCL under Grant PCL2024A06, and the Project of Guangdong Provincial Key Laboratory of Information Security Technology (Grant No. 2023B1212060026).

References

[1] Kasra Arabi, Benjamin Feuer, R Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. *arXiv preprint arXiv:2412.04653*, 2024. 3

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022. 3

[3] Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Transactions on Image Processing (IEEE TIP)*, 10(5):783–791, 2001. 3

[4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 3

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 3

[6] Benedikt Boehm. Stegexpose-a tool for detecting lsb steganography. *arXiv preprint arXiv:1410.6656*, 2014. 5, 7

[7] Zixuan Chen, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Cunerf: Cube-based neural radiance field for zero-shot medical image arbitrary-scale super resolution. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 21185–21195, 2023. 3

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12882–12891, 2022. 3

[9] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3054–3058. IEEE, 2022. 3, 6, 7

[10] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023. 3

[11] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5354–5363, 2024. 1

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 3

[13] 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 *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 33037–33060, 2024. 2, 3, 6, 7

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12087–12097, 2024. 2, 3, 4, 6, 7

[15] Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting watermark for protecting copyrights in radiance fields. *arXiv preprint arXiv:2409.13222*, 2024. 2, 3

[16] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2023. 2

[17] Younghyun Jo, Sejong Yang, and Seon Joo Kim. Investigating loss functions for extreme super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, pages 424–425, 2020. 6

[18] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. In *ACM SIGGRAPH*, pages 165–174. ACM New York, NY, USA, 1984. 3

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ACM TOG)*, 42(4):1–14, 2023. 1, 3, 4, 5, 6, 7

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2

[22] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum (CGF)*, 40 (4), 2021. 3

[23] Martin Kutter, Frederic D Jordan, and Frank Bossen. Digital signature of color images using amplitude modulation. In *Storage and Retrieval for Image and Video Databases V*, pages 518–526. SPIE, 1997. 3

[24] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6526, 2024. 1, 3

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang,

Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–55. Springer, 2024. 2

[26] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13548–13557, 2020. 3

[27] Xiyang Luo, Yinxiao Li, Huiwen Chang, Ce Liu, Peyman Milanfar, and Feng Yang. Dvmark: a deep multiscale framework for video watermarking. *IEEE Transactions on Image Processing (IEEE TIP)*, 2023. 3

[28] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 22401–22411, 2023. 2, 3, 4, 6, 7

[29] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, pages 1532–1542, 2022. 3, 6, 7

[30] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18039–18048, 2024. 3

[31] 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 Transactions on Graphics (ACM TOG)*, 38(4):1–14, 2019. 2, 3, 6, 7, 8

[32] 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. 2, 3, 5, 6, 7, 8, 1

[33] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ACM TOG)*, 41(4):1–15, 2022. 3

[34] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*, pages 271–274. IEEE, 2008. 3, 6

[35] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11453–11464, 2021. 3

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

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Proceedings of the International Conference on Neural Information Processing Systems Workshop (NeurIPS Workshop)*, 2017. 6

[38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–10, 2022. 3

[39] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 253–259, 1984. 3

[40] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 49–56, 1999. 3

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICLR)*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 1

[42] MS Raval and PP Rege. Discrete wavelet transform based multiple watermarking scheme. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, pages 935–938. IEEE, 2003. 3

[43] Christoph Reich, Biplob Debnath, Deep Patel, and Srimat Chakradhar. Differentiable jpeg: The devil is in the details. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4126–4135, 2024. 6

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 3

[45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 36479–36494, 2022. 3

[46] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 20154–20166. Curran Associates, Inc., 2020. 3

[47] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1606–1616, 2024. 3

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–9, 2020. 3

[49] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. Protecting nerfs' copyright via plug-and-play watermarking base model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 57–73. Springer, 2024. 2, 3

[50] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. Geometry cloak: Preventing tgs-based 3d reconstruction from copyrighted images. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 119361–119385, 2024. 3

[51] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10208–10217, 2024. 1, 3

[52] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2126, 2020. 3, 6, 7

[53] Peining Tao and Ahmet M Eskicioglu. A robust multiple watermarking scheme in the discrete wavelet transform domain. In *Internet Multimedia Management Systems V*, pages 133–144. SPIE, 2004. 3

[54] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

[55] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9065–9076, 2023. 3

[56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (IEEE TIP)*, 13(4):600–612, 2004. 7

[57] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 8406–8441, 2023. 3

[58] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 58047–58063, 2023. 3

[59] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19595–19604, 2024. 1, 3

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10031–10040, 2022. 3, 4

[61] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 3

[62] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. 1, 3

[63] Ye Yuan, Xueting Li, Yangyi Huang, Shalini De Mello, Koki Nagano, Jan Kautz, and Umar Iqbal. Gavatar: Animatable 3d gaussian avatars with implicit mesh learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–905, 2024. 1, 3

[64] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 10223–10234, 2020. 3

[65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[66] Xuanyu Zhang, Jiarui Meng, Runyi Li, Zhipei Xu, Jian Zhang, et al. Gs-hider: Hiding messages into 3d gaussian splatting. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 49780–49805, 2024. 2, 3

[67] Yushu Zhang, Jiahao Zhu, Mingfu Xue, Xinpeng Zhang, and Xiaochun Cao. Adaptive 3d mesh steganography based on feature-preserving distortion. *IEEE Transactions on Vision Computer Graphics (IEEE TVCG)*, 30(8):5299–5312, 2024. 3

[68] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, pages 8643–8672, 2024. 7

[69] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 682–697, 2018. 2, 3, 6, 7

[70] Jiahao Zhu, Yushu Zhang, Xinpeng Zhang, and Xiaochun Cao. Gaussian model for 3d mesh steganography. *IEEE Signal Processing Letters (IEEE SPL)*, 28:1729–1733, 2021. 3

[71] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10324–10335, 2024. 3