

MarkSplatter: Generalizable Watermarking for 3D Gaussian Splatting Model via Splatter Image Structure

Xiufeng Huang
xiufenghuang@life.hkbu.edu.hk
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China

Ziyuan Luo
ziyuanluo@life.hkbu.edu.hk
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China

Qi Song
qisong@life.hkbu.edu.hk
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China

Ruofei Wang
ruofei@life.hkbu.edu.hk
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China

Renjie Wan*
renjiewan@hkbu.edu.hk
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China

Abstract

The growing popularity of 3D Gaussian Splatting (3DGS) has intensified the need for effective copyright protection. Current 3DGS watermarking methods rely on computationally expensive fine-tuning procedures for each predefined message. We propose the first generalizable watermarking framework that enables efficient protection of Splatter Image-based 3DGS models through a single forward pass. We introduce GaussianBridge that transforms unstructured 3D Gaussians into Splatter Image format, enabling direct neural processing for arbitrary message embedding. To ensure imperceptibility, we design a Gaussian-Uncertainty-Perceptual heatmap prediction strategy for preserving visual quality. For robust message recovery, we develop a dense segmentation-based extraction mechanism that maintains reliable extraction even when watermarked objects occupy minimal regions in rendered views. Project page: <https://kevinhuangxf.github.io/marksplatter>.

CCS Concepts

• Security and privacy → Digital rights management.

Keywords

Digital watermarking; 3D Gaussian Splatting

ACM Reference Format:

Xiufeng Huang, Ziyuan Luo, Qi Song, Ruofei Wang, and Renjie Wan. 2025. MarkSplatter: Generalizable Watermarking for 3D Gaussian Splatting Model via Splatter Image Structure. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, October 27–31, 2025, Dublin, Ireland. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746027.3758144>

1 Introduction

The remarkable advancements in 3D Gaussian Splatting generative models [12, 37, 39] have intensified the need for effective copyright

*Corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License. *MM '25, Dublin, Ireland*

© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2035-2/2025/10
<https://doi.org/10.1145/3746027.3758144>

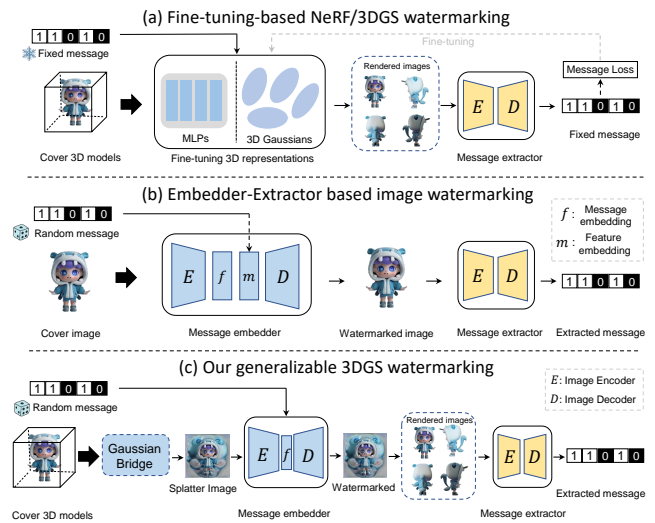


Figure 1: Comparison of different watermarking approaches. (a) Current fine-tuning-based NeRF/3DGS watermarking requires model-specific optimization. (b) Generalizable image watermarking using an embedder-extractor architecture for arbitrary images. (c) Our proposed generalizable 3DGS watermarking framework enables efficient random message embedding for Splatter Image-based 3DGS. By using GaussianBridge as an optional plug-in, our framework can be extended to point cloud-based 3DGS.

protection of 3D assets. However, as we discussed in **Figure 1 (a)**, in current watermarking settings for 3DGS [50, 51], existing methods rely on computationally expensive fine-tuning procedures for each predefined message. Such a per-scene-specific setting can only support fixed watermark embedding. Once the watermark is embedded, it cannot be dynamically modified without repeating the entire fine-tuning process, severely limiting practical deployment in real-world applications.

A more practical scenario should be a generalizable watermarking pipeline that enables on-demand watermark embedding into

pre-trained 3DGS models without requiring repeated optimization procedures. This framework would allow for seamless embedding of arbitrary messages through a single forward pass, facilitating efficient adaptation to different copyright messages and 3D contents while preserving rendering quality and maintaining robustness.

The effectiveness of such generalizable frameworks has been well-established in image watermarking [32, 63, 64] as shown in **Figure 1 (b)**. In current settings for image watermarking, neural networks process images to embed and extract watermarks in a single forward pass, enabling rapid implementation of copyright protection across different contents. However, directly adapting such approaches for 3DGS models encounters significant challenges. **First**, the absence of a framework that can directly process 3DGS representations limits the development of efficient watermarking solutions. Unlike 2D images with regular grid structures, 3DGS models consist of unstructured point cloud-based 3D Gaussians with varying attributes, making it non-trivial to apply existing neural network architectures for watermark embedding and extraction. **Second**, existing image watermark extractors [63, 64] demonstrate limited effectiveness when processing rendered images from 3D models. This limitation is particularly evident when watermarked content only occupies a small portion of the rendered image due to camera zoom-out or perspective changes, a common scenario in 3D visualization. Traditional extractors [63, 64], which decode watermarks globally from the entire image, often fail to accurately extract messages when the watermarked object appears at different scales and positions. Without addressing these challenges, current watermarking methods rely on computationally intensive optimization procedures that lack the flexibility and efficiency needed for practical deployment.

To address these challenges, we propose a novel framework for generalizable watermarking of 3D Gaussian Splatting models [17] as illustrated in **Figure 1 (c)**. We identify a new structure for 3DGS, aka the Splatter Image [37]. It is a grid-based 3DGS format, which maintains the same information from the point cloud-based 3DGS model. By flattening the 2D grid-based 3DGS data into the 1D unstructured format, the Splatter Image representation can be seamlessly converted to the native point cloud-based 3DGS structure. More importantly, such a format is compatible with conventional neural networks, which enables direct processing through standard neural architectures [6, 10]. The data owners can directly utilize trained neural networks for message embedding and extraction, just like the established convenient settings in image watermarking [63, 64].

We design an **embedder-extractor** watermarking framework for 3DGS message embedding and extraction, similar to the proven convenient setup in image watermarking [14, 64]. Our message embedder utilizes an encoder-decoder architecture that jointly optimizes two complementary mechanisms: in-attribute embedding for encoding messages within Splatter Image’s color attributes, and cross-attribute constraints to regulate perturbation magnitudes. These perturbations are adaptively modulated through a Gaussian Uncertainty Perceptual (GUP) heatmap, which constrains attributes with high perceptual sensitivity to make the embedded watermarks **imperceptible**.

For message extraction, we propose a dense segmentation approach that treats watermark extraction as a pixel-wise message

localization and prediction task, ensuring **robust** extraction even when watermarked regions occupy minimal image space. This segmentation-based mechanism overcomes the limitations of traditional extractors that rely on the global watermark embedding and maintains effectiveness across various viewing conditions.

With many point cloud-based 3DGS models already in existence, we introduce GaussianBridge, a plug-and-play module that transforms unstructured point cloud-based 3DGS models into Splatter Image format. Leveraging a Gaussian reconstruction module as the core, we devise a method to fully capture point cloud information. This promotes the protection for both Splatter Image and point cloud-based 3DGS models. Upon completion of the watermarking procedure on Splatter Image, the watermarked 3DGS models can also be easily converted back to the point cloud structure by the flattening operation.

In summary, our contributions are listed below:

- We propose the first generalizable 3DGS watermarking framework that operates directly on 3D Gaussians via the Splatter Image structure, eliminating the need for computationally expensive per-scene optimization while enabling scalable protection across diverse 3DGS models.
- We introduce GaussianBridge, a novel conversion module that seamlessly transforms 3DGS models from their native point cloud format to the Splatter Image representation, enabling unified watermarking protection for existing point cloud-based 3DGS architectures.
- We introduce a novel embedder-extractor architecture that achieves the first unified neural network framework for both image and 3DGS watermarking, effectively addressing the domain gap between 2D and 3D watermarking via the Splatter Image structure and GaussianBridge module.

We call our solution **MarkSplatter**. With MarkSplatter, data owners can apply our generalizable 3DGS watermarking to protect newly created Splatter Image-based 3DGS models [13, 39, 52] directly. They can also effortlessly transform point cloud-based 3DGS models into the Splatter Image structure. MarkSplatter enhances the security of 3DGS model distribution. Extensive experiments demonstrate that our method achieves advanced performance in both reconstruction quality and watermark robustness.

2 Related work

3D Gaussian Splatting. 3DGS [17] has been rapidly adopted across multiple domains and demonstrated remarkable 3D reconstruction results. There is a growing interest in reconstructing 3D objects from sparse inputs for generative tasks [9, 22, 28, 56], following two main approaches: per-scene optimization methods and feed-forward inference methods. The former methods typically leverage multi-view geometry constraints to jointly optimize rendering results and camera poses [5, 29, 41]. Feed-forward inference methods [3, 45, 56] can reconstruct entire scenes in a single pass without additional optimization. Splatter Image [37] is the first method to reconstruct entire 3DGS model from just one input image, which can convert 3DGS models from unstructured point-cloud format into a structured image-like representation. Recent methods like LGM [39], GRM [52], and GS-LRM [60] have further developed this approach using various architectures to reconstruct Gaussian

features, making them compatible with image foundation models [6].

Image watermarking. Traditional image watermarking approaches embed information either in spatial domains [42] or frequency domains using techniques such as Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) [20, 30]. The advent of deep learning has led to significant advances in image watermarking, achieving both high quality and robustness [38, 46–48, 54, 58, 61]. HiDDeN [64], a pioneering deep learning-based method, demonstrated superior performance compared to traditional approaches. Subsequent research has enhanced deep learning-based watermarking through various techniques, such as scalable residual connections [1], attention mechanisms [57, 59], and invertible networks [8, 26]. Recently, the Watermark Anything Model (WAM) [32] reformulates watermark extraction as a segmentation task, enabling pixel-level detection and reliable message extraction from partial image regions. This makes WAM especially effective for images with white or transparent backgrounds, such as rendered 3D objects.

3D watermarking. 3D watermarking can differ significantly from image watermarking. Traditional 3D watermarking approaches are designed for explicit 3D models [2, 31, 34, 49, 55]. Recent works [23–25, 35, 36, 50] have shown great progress on neural rendering-based 3D models, including Neural Radiance Fields (NeRF) and 3DGS. CopyRNeRF [23] introduces watermarked color representations for invisible NeRF protection with high rendering quality. WateRF [14] employs DWT-based watermarking in NeRF space with patch-wise optimization for better robustness. GaussianMarker [51] embeds watermarks into 3DGS through uncertainty-based perturbations for high-quality protection. However, these NeRF and 3DGS watermarking approaches are all fine-tuning-based methods, requiring post-optimization that takes dozens of minutes or even hours. This motivates us to develop generalizable 3DGS watermarking in a feed-forward inference manner, which takes only seconds for any 3DGS model, making it practical for real-world applications.

3 Preliminary

3DGS. 3DGS [17] models the 3D scenes or objects as a collection of 3D Gaussians. Each 3D Gaussian is parameterized by its mean $\mu \in \mathbb{R}^3$ as the position, opacity $\alpha \in \mathbb{R}^3$ for transparency, scaling S and rotation R as the shape, and spherical harmonics $c \in \mathbb{R}^{3 \times d}$ as the view-dependent color. In our experiments, we focus on view-independent RGB color $S \in \mathbb{R}^3$ in the spherical harmonics for each Gaussian. During rendering, 3DGS follows a typical neural point-based approach [19] to compute the color C of a pixel by blending N depth ordered points.

Splatter Images. While 3DGS achieves remarkable speed and visual quality, its unstructured, permutation-invariant representation poses fundamental challenges for neural network processing. Unlike conventional grid-based data such as images, 3DGS roots in point cloud structure, which complicates its compatibility with established image-foundation neural networks [6, 10]. To address this, recent works [39, 52] propose generating 3D Gaussians from multi-view images into **Splatter Images**, a structured $H \times W$ grid representation where each pixel corresponds to a 3D Gaussian parameterized by 14 channels. These channels encode attributes for

position μ , color c , opacity σ , rotation r , and scaling s , effectively modeling both appearance and geometry as spatially aligned image-like tensors. By regularizing unstructured 3D Gaussians into this grid format, Splatter Images bridge the critical gap between 2D images and 3D Gaussians, enabling the application of generalizable watermarking for 3DGS.

4 Proposed method

Our generalizable watermarking for 3DGS contains two key components: (1) The **GaussianBridge** module to convert the unstructured 3DGS model into Splatter Images, a grid-based 3DGS representation. (2) The **watermarking model** to embed copyright messages on the reconstructed Splatter Images and extract the embedded message from the rendered images.

4.1 GaussianBridge

Our whole approach is based on the Splatter Images to achieve generalizable watermarking of 3DGS. However, with many point cloud 3DGS models in existence, we still need to consider their protection. We thus introduce the GaussianBridge module with a plug-and-play property that establishes a conversion pathway between point cloud structures and grid-based Splatter Image structures. Then, the Splatter Images can be directly processed by the neural networks to achieve direct and generalizable watermarking.

As shown in Figure 2, our GaussianBridge renders multi-view images orbitally around the 3DGS object with their camera parameters. Then, a Gaussian reconstruction model is utilized to transfer the 3DGS models from 3D point cloud formats into grid-based Splatter Images representations. To comprehensively preserve the information in the original 3DGS model with point-cloud structures, we propose a strategy to change the input setting of the current Gaussian reconstruction from a fixed number of images to an arbitrary number of images.

Since the GaussianBridge module leverages the Splatter Images structure to transform 3D Gaussian Splatting models, our method can be applied to per-pixel-based Gaussian reconstruction models. Specifically, we adopt LGM [39] as the baseline method for our Gaussian reconstruction model due to its strong performance and balance between reconstruction quality and computational efficiency. Furthermore, our method can also be extended to more advanced Gaussian reconstruction models such as GRM and StereoGS, which enables our approach to be well-suited for both 3D Gaussian reconstruction tasks and 3D Gaussian generation tasks.

Specifically, we adapt the LGM [39] baseline from a fixed number of input views setting to handle arbitrary input views to achieve high-quality 3DGS reconstructions. We first densely encode camera poses using Plücker ray embedding [53], then concatenate multi-view images and ray embeddings to form a unified 9-channel feature map for 3D Gaussian prediction. This multi-view feature embedding F is defined as: $F = \{C, R_o \times R_d, R_d\}$, where C is the images color and R_o, R_d represent ray origins and directions, respectively.

The asymmetric U-Net architecture in LGM takes F as input and employs residual layers [10] and self-attention mechanisms [43], following designs from prior works [11, 27, 37]. To enhance cross-view information propagation, we flatten and concatenate multi-view features before self-attention layers, aligning with strategies in

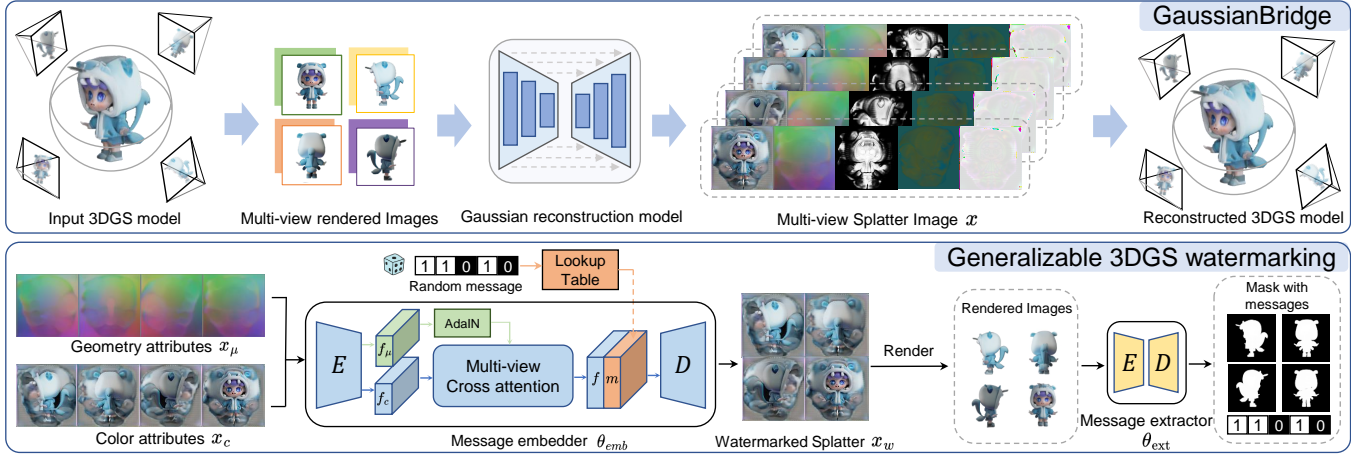


Figure 2: Overview of our proposed method. The GaussianBridge module enables bi-directional transformation between 3D Gaussian Splatting (3DGS) models and Splatter Images. We propose novel 3DGS generalizable watermarking through a two-phase approach: a message embedder that applies a selective strategy to introduce targeted perturbations in the Splatter Image, and a message extractor that generates segmentation masks to precisely locate watermarked regions and retrieve the embedded messages.

multi-view diffusion models [33, 44]. The final output feature map comprises 14 channels, with each pixel representing a 3D Gaussian parameterized by position μ , scaling s , rotation r , opacity σ and color c . This structured representation aligns with the SplatterImage format [37], ensuring compatibility with generalizable watermarking.

4.2 Generalizable 3DGS watermarking

We introduce a generalizable watermarking system capable of directly watermarking 3DGS models. Unlike previous approaches that require computationally expensive fine-tuning and optimization processes, our method offers an efficient alternative for 3DGS protection.

As illustrated in Figure 2, our proposed framework consists of a watermark embedder θ_{emb} and a watermark extractor θ_{ext} . The embedder θ_{emb} generates a watermarked Splatter Image x_w by adding a perturbation δ on the original Splatter Image x . The watermark extractor θ_{ext} predicts a segmentation mask y^{mask} to localize watermarked regions and extracts pixel-wise embedded messages y^{msg} from watermarked rendered images.

4.2.1 Multi-view watermark embedder. Our model accepts multi-view Splatter Images x and a binary message \hat{m} as input. Following proven image watermarking paradigms [32, 64], we adopt an encoder-decoder architecture for the embedder. However, 3DGS models introduce unique challenges due to their multi-attribute structure. To ensure watermark imperceptibility, we must consider both **in-attribute consistency** and **cross-attribute correlations**. We propose a dual-strategy approach: *in-attribute embedding* to encode messages within selective attributes, and *cross-attribute constraint* to modulate perturbations across attributes. Such an approach leverages the inherent structure of 3DGS to achieve imperceptibility.

In-attribute embedding. We apply the embedder on the Splatter Image color attribute x_c to generate watermarked color attributes, and keep other attributes unchanged. This selective embedding strategy ensures the efficient and effective message embedding while avoiding perturbing the 3DGS geometry [23]. The embedder takes x_c as the input and the position attribute x_μ as geometrical guidance to produce watermarked color attributes that harmonize with the underlying 3D geometry.

The embedder first encodes x_c and x_μ with ResidualNet-based [10] down-sampling networks into feature representations f_c and f_μ . To ensure geometric consistency, we apply adaptive instance normalization (AdaIN) layers to dynamically modulate these color features f_c using encoded geometry attributes f_μ , enabling joint reasoning between appearance and geometry. This modulated feature representation is further refined through a multi-view self-attention mechanism, which explicitly models inter-view dependencies and ensures the learned representation adheres to the 3DGS multi-view geometric constraints [33, 39]:

$$f = \text{MultiViewAttn}(\text{AdaIN}(f_c, f_\mu)), \quad (1)$$

where f is the learned middle representation. Message embedding occurs at the bottleneck layer, where a learnable binary lookup table maps binary messages into latent perturbations. These perturbations are concatenated with the features f , ensuring the watermark embedding process is established on both appearance and geometric priors of the 3DGS model. The decoder then processes the concatenated representation through additional multi-view self-attention layers and ResidualNet-based [10] up-sampling networks to generate a perturbation map $\delta = \theta_{emb}(x_c, \hat{m})$. This perturbation is applied to the original color attribute x_c to produce the final watermarked color attributes x_{c_w} , ensuring minimal perceptual distortion while embedding recoverable information:

$$x_{c_w} = x_c + \theta_{emb}(x_c, \hat{m}). \quad (2)$$

The watermarked attribute x_{c_w} replaces x_c in the original Splatter Image x , producing the watermarked version x_w .

Cross-attribute constrains. To ensure the imperceptibility, we propose Gaussian Uncertainty Perceptual (GUP) heatmap, which utilizes uncertainty to constrain the across-attributes interferences caused by the embedded watermarks.

In previous fine-tuning-based 3D watermarking [15, 21], many methods rely on reference-based methods to estimate the rendering contribution map or saliency map to constrain perturbation. These methods rely on the ground truth data, which is impractical in the inference situations for **3DGS generative models** [39, 52, 60] when ground truth images are unavailable. While uncertainty-based estimation [51] provides a non-reference-based approach by adding perturbations into the model parameters, which is suitable for the 3DGS generative models without referencing ground truth images. Thus, we use the uncertainty estimation methods [16, 51] for constraining watermark perturbation. We use the simplification of the Hessian matrix \mathbf{H} as the approximated Fisher information to estimate uncertainty values \mathbf{U} for each Gaussian [18, 51]: $\mathbf{H} = \nabla_{\mathcal{G}} I_{\mathcal{G}} \nabla_{\mathcal{G}} I_{\mathcal{G}}^T$, where $\nabla_{\mathcal{G}}$ is the gradient for the parameters of 3DGS model \mathcal{G} and I is the rendered image. We estimate \mathbf{U} by computing logarithm of \mathbf{H} for rearrange the exponential large values:

$$\mathbf{U} = \log(\nabla_{\mathcal{G}} I_{\mathcal{G}} \nabla_{\mathcal{G}} I_{\mathcal{G}}^T). \quad (3)$$

Specifically, we exclude rotations r from \mathcal{G} because the changes in this attribute can easily cause noticeable differences such as needling artifacts on the boundary regions.

Given the uncertainty values \mathbf{U} , we can generate GUP heatmaps: $\gamma = \mathcal{R}(\mathbf{U}, v)$, where \mathcal{R} is the splatting rendering function [16] and v is a normalized viewing camera parameters for Splatter Images. These GUP heatmaps γ are modulated on Splatter Images to constrain perturbations, thus ensuring imperceptible watermarking:

$$x_w = x + \alpha \cdot \gamma \odot \delta, \quad (4)$$

where γ is the GUP heatmaps to modulate the perturbation δ , and α is the scaling factor to adapt δ .

4.2.2 Robust watermark extractor. We design our watermark extractor as a segmentation-based approach that simultaneously localizes watermarked regions and retrieves embedded messages. Following the watermark-anything model (WAM) [32], our architecture combines a Vision Transformer (ViT) [6] encoder for global feature extraction and a pixel decoder that upsamples features to match the input resolution, producing per-pixel outputs $y = \{y^{mask}, y^{msg}\}$ of size $1 + N$ bits. This design eliminates the need for an auxiliary discriminator [21, 50].

Previous methods [14, 15, 50, 51] often rely on HiDDeN’s extractor [64], which struggle to accurately retrieve watermarks when applied to 3D models, as watermarked pixels frequently occupy sparse regions under varying camera viewpoints. Our watermark extractor θ_{ext} localizes and extracts embedded messages from watermarked rendered images I_w using:

$$I_w = \mathcal{R}(x_w, V); y^{mask}, y^{msg} = \theta_{ext}(I_w) \quad (5)$$

where \mathcal{R} represents the rendering function, V denotes the viewing camera, y^{mask} is the segmentation mask that localizes watermarked regions, and y^{msg} represents the pixel-wise embedded messages.

Furthermore, most 3D watermarking methods [14, 23] cannot verify copyright messages across both model parameters and rendered images. Our method extracts watermarks from 3DGS models via Splatter Images without scene-specific tuning. Specifically, the watermark embedded in the watermarked Splatter Image color attribute x_c can be directly extracted by our message extractor θ_{ext} , thus ensuring the generalizability for embedding and extracting from 3DGS model parameters.

Augmentation. Our augmentation pipeline simulates real-world distortions on 2D rendered images via geometric transformations (resize, scaling, rotation, translation, perspective, crop-out) and photometric adjustments (noise, JPEG compression, blur, brightness/contrast shifts), reflecting typical degradations on real-world applications to obtain the watermarking robustness. To ensure 3D robustness, we also introduce 3D-aware augmentations (geometric edits: cropping/translation/rotation; noise injection) on Splatter Images, preserving extraction accuracy even when malicious alter the 3D representations.

4.3 Training losses

The training minimizes the objective function \mathcal{L}_{total} which is a linear combination of the mask and message losses: $\mathcal{L}_{total} = \lambda_1 \cdot \mathcal{L}_{mask} + \lambda_2 \cdot \mathcal{L}_{msg}$. The mask loss is the average of the pixel-wise cross-entropy between detecting mask y^{mask} and the ground truth \hat{y}^{mask} to predict whether a pixel contains watermark message or not.

$$\mathcal{L}_{mask} = -(y^{mask} \log(\hat{y}^{mask}) + (1 - y^{mask}) \log(1 - \hat{y}^{mask})). \quad (6)$$

Similarly, the message loss is the average of the pixel-wise and bit-wise binary cross-entropy between the predicted watermark messages y^{msg} and the ground truth watermark messages \hat{m} :

$$\mathcal{L}_{msg} = -(y^{msg} \log(\hat{m}) + (1 - y^{msg}) \log(1 - \hat{m})). \quad (7)$$

5 Experiments

5.1 Implementation details

Dataset. We choose the Objaverse dataset [4] as our training dataset to train our watermarking models. Specifically, we utilize the LVIS annotation to obtain a subset of Objaverse [4], which contains around 45k high-quality objects. For each object in the dataset, we render 512×512 images from 32 random viewpoints. The 32 rendered views from different angles are then used to train 3DGS models for 10K iterations using [17]. This way, we create a high-quality and large-scale 3DGS dataset of 45K objects and 144k rendering views from Objaverse [4], which is sufficient to train our GaussianBridge module and watermarking models. We evaluate the qualitative and quantitative results for reconstruction and watermarking performance using Objaverse [4] and Google Scanned Objects (GSO) [7] datasets. For the Objaverse dataset, we randomly sample 200 objects from the LVIS annotation as evaluation dataset and others as training dataset. The GSO dataset comprises around 1,000 objects, from which we randomly select 50 objects as our evaluation dataset. Specifically, we render 16 images of each object in the GSO evaluation set in an orbiting trajectory with uniform azimuths varying positive elevations in $\{0^\circ, 20^\circ\}$ for sampling on the top semi-sphere of an object.


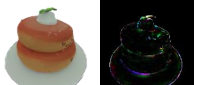





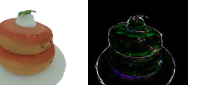

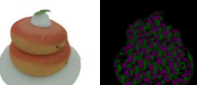


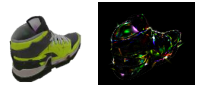

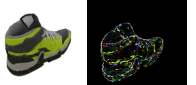

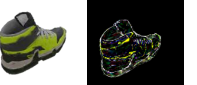
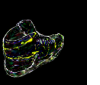
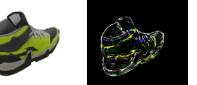

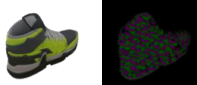






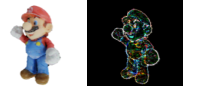

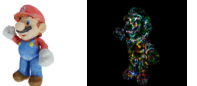








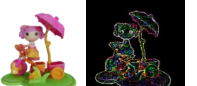
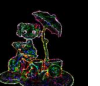




Ground Truth	HiDDeN + 3DGS		WAM + 3DGS		WaterRF + 3DGS		GaussianMarker		Ours	
	Watermarked	Difference (x10)	Watermarked	Difference (x10)	Watermarked	Difference (x10)	Watermarked	Difference (x10)	Watermarked	Difference (x10)
 PSNR/ACC/TIME(s)										
	34.89/68.75%/239.6		35.63/52.41%/297.9		34.19/90.41%/141.7		35.38/89.16%/107.6		36.25/94.53%/2.6689	
 PSNR/ACC/TIME(s)										
	33.47/60.41%/241.4		31.91/53.12%/328.2		32.49/89.58%/148.1		33.28/91.25%/113.4		33.82/92.96%/2.5106	
 PSNR/ACC/TIME(s)										
	35.77/64.58%/265.8		35.53/51.04%/343.7		32.40/90.70%/195.2		32.34/93.40%/126.0		36.74/95.31%/2.7889	
 PSNR/ACC/TIME(s)										
	32.48/68.75%/275.1		31.67/51.56%/374.2		32.12/92.70%/185.7		31.82/92.70%/121.7		32.71/93.75%/2.2527	

Figure 3: Qualitative results of comparing our method with the baselines on the evaluation dataset. The extraction accuracies are reported in 32 bits. The differences between watermarked and ground truth images are amplified by 10 times for better visualization.

Training settings. Our training contains two stages, and all experiments are trained on 8 NVIDIA V100 (32G) GPUs for about 1 day. In the first stage, we train our GaussianBridge module to obtain the high-quality multi-view Splatter Image representations. We use LGM [39] as the baseline for 3D Gaussian reconstruction model and use 4 images as the default number of views and enabling to increase in the number of views up to 8. The size of the reconstructed Splatter Image is 128×128 for each input view, and the 3D Gaussians are rendered at 512×512 resolution.

In the second stage, we utilize the GaussianBridge module for obtaining multi-view Splatter Images from the 3DGS models in the training dataset to train our watermarking models. For any 3DGS model, we render 4 images with camera elevation angle = 0 and azimuths degrees = {0, 90, 180, 270} to cover the entire object. The GaussianBridge module uses the rendered images to reconstruct multi-view Splatter Images. We apply our multi-view watermark embedder on the Splatter Images and then randomly select 8 camera views to render watermarked novel views for message extraction. We set the training batch size = 4 and use AdamW optimizer for the multi-view watermark embedder. The training epoch is set to 30 epochs. The learning rate is linearly increased from 10^{-6} to 10^{-5} over the first 5 epochs, and then follows a cosine schedule down to 10^{-7} until epoch 30. We apply the augmentation pipeline to ensure robustness during training. We also extract messages from the watermarked Splatter Images to ensure robustness in both 2D rendered images and 3D Gaussians parameters.

5.2 Experimental settings

Baselines. We design experiments to validate the message extraction on both rendered 2D images and 3D Gaussian parameters, demonstrating the effectiveness of our proposed method. For message extraction from rendered images, we compare our proposed

method with four baselines for a fair comparison: 1) **Fine-tuning 3DGS+HiDDeN** [64]: Preprocessing images with the classical image watermarking method HiDDeN [64] before the training of 3DGS [17]; 2) **Fine-tuning 3DGS+WAM** [64]: Preprocessing images with the state-of-the-art image watermarking method HiDDeN [32] before the training of 3DGS [17]; 3) **3DGS+WaterRF**: Train a watermarked 3DGS object using frequency-based watermarking method via WaterRF [14] approach; 4) **GaussianMarker**: state-of-the-art fine-tuning-based 3DGS watermarking method via uncertainty estimation; For 3D message extraction, we compare our method with the GaussianMarker baseline since other methods do not provide a 3D watermarking method. To illustrate our method can be generalizable to different 3D Gaussian generative models, we select three baselines, including DreamGaussian [40], Triplane Gaussians [65] and LGM [39]. To ensure a fair comparison, we first build the point cloud 3DGS model similar to the baselines. Then, we use our pre-trained GaussianBridge to convert 3DGS from the point cloud to Splatter Images.

Evaluation methodology. We set the bit length of binary messages to 32 and 48 bits to test *capacity* for all baselines and our method. We evaluate the performance of our proposed method by comparing it with other digital watermarking baselines using the standard of imperceptibility and robustness. For *imperceptibility*, we evaluate the reconstruction quality with PSNR, SSIM, and LPIPS [62] for rendered images. For *robustness*, we evaluate whether the binary messages in rendered images can remain consistent against various distortions, including 2D Gaussian noise with standard deviation $v = 0.1$, JPEG compression with quality factor $Q = 40$, scaling with the size factor $sc \leq 70\%$, and Gaussian blur with kernel size $k = 3$ and the standard deviation $\xi = 0.1$. We also evaluate whether the copyright messages in 3D Gaussians can remain consistent against various 3D attacks, including 3D Gaussian noise

Table 1: Reconstruction qualities and bit accuracy compared with different baselines. PSNR, SSIM and LPIPS are computed between the watermarked images and ground truth images. The results are computed on the average of samples from Objaverse and GSO datasets.

Dataset	Method	32 bit					48 bit				
		Bit Acc \uparrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Time(s) \downarrow	Bit Acc \uparrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Time(s) \downarrow
Objaverse + GSO	HiDDeN + 3DGS	66.03	33.71	0.9022	0.0782	255.34	63.19	32.87	0.9041	0.0995	269.86
	WAM + 3DGS	52.03	33.95	0.8929	0.0801	336.61	50.84	32.82	0.8896	0.1005	347.53
	WaterF + 3DGS	90.84	32.63	0.8973	0.0951	167.61	90.14	31.40	0.8716	0.0939	175.35
	GaussianMarker	91.62	33.20	0.9082	0.0822	117.14	90.93	32.69	0.8909	0.0965	126.08
	Ours	94.41	35.12	0.9234	0.0775	2.53	93.36	33.99	0.9189	0.0891	2.65

Table 2: Quantitative assessment of robustness against various attacks, compared to baseline methods. The reported results are averaged across the Objaverse and GSO datasets. All experiments were conducted using 32-bit messages. The results are computed on the average of all examples.

Dataset	Method	Bit accuracy \uparrow (%)							
		None	Noise ($\nu = 0.1$)	JPEG ($Q = 40$)	Scaling ($sc \leq 70\%$)	Blur ($\xi = 0.1$)	Rotation ($ro \leq \pm\pi/6$)	Translation ($t \leq 20\%$)	Crop-out ($cr \leq 50\%$)
Objaverse + GSO	HiDDeN + 3DGS	66.03	64.71	53.34	50.86	65.24	59.15	64.83	52.17
	WaterF + 3DGS	90.84	86.23	75.80	58.08	89.64	85.42	89.43	57.67
	GaussianMarker	91.62	90.98	71.66	61.50	87.75	81.32	87.75	59.15
	Ours	94.35	93.20	89.06	89.95	92.31	92.45	91.88	87.08

with standard deviation $\nu' = 0.1$, translation with the ratio factor $t \leq 20\%$, rotation with the angular factor $ro \leq \pm\pi/6$, and crop-out with the cropping size percentage $cr \leq 50\%$.

5.3 Experimental results

Qualitative results. We use PSNR, SSIM, and LPIPS [62] to measure the Gaussian reconstruction quality. Figure 3 demonstrates the qualitative and quantitative reconstruction results on the Objaverse and GSO datasets. We compare the reconstruction qualities and bit accuracies with all baselines, and the qualitative results are shown in Figure 3. Both **HiDDeN + 3DGS** and **WAM + 3DGS** watermark the training images but struggle to decode the watermark messages, which align with previous methods [23, 51] since it is difficult to transmit the watermark signal from the 2D images into the 3DGS models. Both **WaterF + 3DGS** and **GaussianMarker** are all fine-tuning-based methods. Although these methods show promising decoding accuracy, they require per-object optimization for more than 100 seconds and also exhibit vulnerability when subjected to zoomed-in/out operations, making them impractical for real-world applications.

Quantitative results. We display the quantitative results in Table 1 and Table 2. Our method shows superior performance on both reconstruction quality and decoding accuracy since we consider the multi-view corresponding and the 3D Gaussian geometry condition. Even with different distortions to the rendered images, our method can still achieve high decoding accuracy to reliably safeguard the 3DGS models. MarkSplatter can also extract the message from the 3D Gaussian attribute via the format of Splatter Image. We compare our method with the GaussianMarker[51] method in Table 3. GaussianMarker [51] relies on a per-scene specific decoder for the

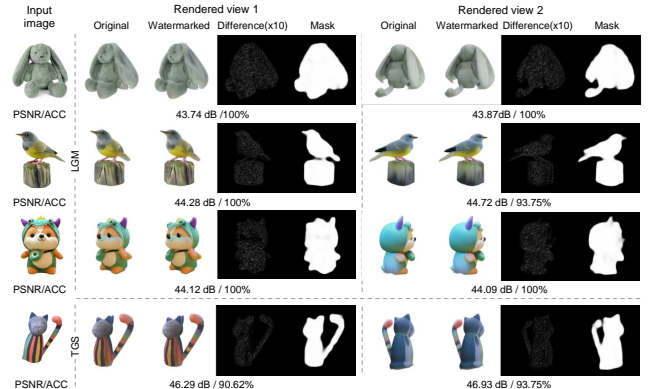


Figure 4: Watermarking performance on generative models. PSNR is computed between the original and watermarked rendered images. The results are averaged across the given scenes and bit accuracy is in 32 bits.

3D watermark decoding, which is impractical for real-world applications. Our MarkSplatter can extract watermark in 3D Gaussians in a generalizable method, which shows superior performance than the fine-tuning methods.

Single-Image-to-3D results. We evaluate our MarkSplatter with the Single-Image-to-3D generative models [39, 65] in Figure 4. In the first column, we display the input single image. In the second and third columns, we show the original generated image, the watermarked image, the difference, and the predicted mask for two different viewing angles. Specifically, in the first 3 rows, we use the LGM [39] pipeline to achieve Single-Image-to-3D, and MarkSplatter can directly apply to the LGM [39] model since its Splatter

Image Structure. In the fourth row, we use the TriplaneGaussian to reconstruct a point cloud-based 3DGS model and use our GaussianBridge to convert it into a Splatter Image-based 3DGS model for MarkSplatter watermarking protection. These results showcase the generalizability of MarkSplatter to protect any 3DGS models in point-cloud format and Splatter Image format.

Table 3: Message extraction from 3DGS model parameters. The results are in 32 bits and computed on the average of samples from Objaverse [4] and GSO [7] datasets.

Method	None	Noise	Rotation	Crop-out	Time(s)
GaussianMarker	100%	96.48%	95.44%	58.28%	680.64
Ours	100%	98.66%	97.70%	93.23%	2.35

5.4 Ablation study

Model components. We conduct a series of ablation studies to evaluate the contributions of individual components in our full model, as detailed in Table 4. Our GUP heatmap ensures imperceptible watermarks in Splatter Images. Removing the heatmap slightly boosts bit accuracy (as unrestricted embedding simplifies message extraction), but severely degrades image quality by making watermarks visible. The GUP heatmap balances this trade-off for watermark embedding to preserve visual quality while maintaining high decoding accuracy. We ablate the cross-attention module and the AdaIN module and experimental results shows degraded performance, confirming their essential roles in guaranteeing message extraction accuracy by integrating spatial information across different viewpoints. Our experiments demonstrate that the full model, incorporating all components, achieves the most comprehensive and robust performance compared to the ablated variants.

Table 4: Ablation study for model components. The results are in 32 bits and computed on the average of samples from Objaverse [4] and GSO [7] datasets.

Setting	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Bit Acc \uparrow
full	35.12	0.9234	0.0775	94.41%
w/o GUP	34.09	0.9132	0.1053	95.12%
w/o cross-attention	32.42	0.8961	0.0939	85.25%
w/o AdaIN	33.30	0.9044	0.0914	82.12%

Number of Splatter Image. To enhance watermark robustness, we investigate increasing the number of input Splatter Images for watermark embedding. While our baseline uses 4 views, we increase the rendered images for GaussianBridge module to generate 6 Splatter Images, which can improve novel-view rendering while maintaining bit accuracy. To train our model for 6 input views, we fine-tune our GaussianBridge module previously trained with 4 input view images. During training, we still randomly select 8 camera views for each object and set the first 6 images as the input images. Since the larger number of input view images, each GPU can use a batch size of 4, resulting in a total batch size of 32. The output feature map is still at the size of 128×128 for each input

view, resulting in output 3D Gaussians to $128 \times 128 \times 6 = 98304$ in total. We use the same learning rate setting for training 4 input views, and the training can be finished in 20 epochs.

We discuss both qualitative and quantitative results when increasing the input number of views from 4 images to 6 images. For the 4-input-views condition, we maintain the same camera pose setting with elevations of 0 and azimuths at $[0, 90, 180, 270]$ degrees. For the 6-input-views condition, we set the camera pose with elevations of 0 and azimuths of $[0, 60, 120, 180, 240, 300]$ degrees. The reconstruction quality is still evaluated at azimuths of $[45, 135, 225, 315]$.

We demonstrate qualitative results in Figure 5. With accurate camera poses and consistent input images, both our models for 4-input-views and 6-input-views can faithfully reconstruct the 3D objects. Moreover, by leveraging six input views with denser azimuth coverage, our model achieves enhanced 3D reconstruction quality. As evidenced by the quantitative results in Table 5, this multi-view configuration yields a marked improvement in rendering quality metrics and an enhancement in bit recovery accuracy. These findings demonstrate that increasing observational viewpoints can strengthen the geometric consistency of 3DGS reconstructions to improve watermark extraction robustness.

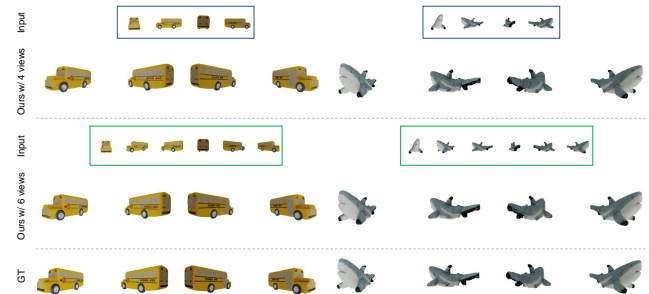


Figure 5: GaussianBridge reconstruction with 4 input views and 6 input views.

Table 5: Evaluation for Gaussian reconstruction and watermarking quality on Objaverse and GSO datasets [7].

Methods	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Acc \uparrow
Ours w/ 4 views	35.12	0.9234	0.0775	94.41
Ours w/ 6 views	36.20	0.9331	0.0741	94.80

6 Conclusion

We propose MarkSplatter, the first generalizable 3DGS watermarking framework for 3DGS models through our GaussianBridge module to encode 3DGS models into compact Splatter Images with high reconstruction quality. A multi-view watermark embedder hides messages in Splatter Images, using a GUP heatmap to limit visible changes, and a robust extractor then locates and reads the messages. This pipeline establishes a foundation for generalizable 3DGS watermarking without compromising reconstruction quality.

Acknowledgement

This work was carried out at the Renjie Group, Hong Kong Baptist University. Renjie Group is supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2022A1515110692, 2024A1515012822.

References

- [1] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. 2020. ReDMark: Framework for Residual Diffusion Watermarking based on Deep Networks. *Expert Systems with Applications* (2020).
- [2] Xingyu Chen, Yu Deng, and Baoyuan Wang. 2023. Mimic3D: Thriving 3D-Aware GANs via 3D-to-2D Imitation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [3] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. 2024. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627* (2024).
- [4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [5] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. 2022. Depth-supervised nerf: Fewer views and faster training for free. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiuhua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations (ICLR)* (2021).
- [7] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. 2022. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*.
- [8] Han Fang, Yupeng Qiu, Kejiang Chen, Jiye Zhang, Weiming Zhang, and Eechien Chang. 2023. Flow-based robust watermarking with invertible noise layer for black-box distortions. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 5054–5061.
- [9] Shuai Guo, Qiuwen Wang, Yijie Gao, Rong Xie, and Li Song. 2024. Depth-Guided Robust and Fast Point Cloud Fusion NeRF for Sparse Input Views. In *The Conference on Artificial Intelligence (AAAI)*.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems NeurIPS* (2020).
- [12] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2023. LRM: Large Reconstruction Model for Single Image to 3D. In *International Conference on Learning Representations (ICLR)*.
- [13] Xiufeng Huang, Ka Chun Cheung, Runmin Cong, Simon See, and Renjie Wan. 2025. Stereo-GS: Multi-View Stereo Vision Model for Generalizable 3D Gaussian Splatting Reconstruction. *arXiv preprint arXiv:2507.14921* (2025).
- [14] Youngdong Jang, Dong In Lee, MinHyuk Jang, Jong Wook Kim, Feng Yang, and Sangpil Kim. 2024. WaterF: Robust Watermarks in Radiance Fields for Protection of Copyrights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12087–12097.
- [15] Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 2024. 3d-gsw: 3d gaussian splatting watermark for protecting copyrights in radiance fields. *arXiv preprint arXiv:2409.13222* (2024).
- [16] Wen Jiang, Boshu Lei, and Kostas Daniilidis. 2024. FisherRF: Active View Selection and Uncertainty Quantification for Radiance Fields using Fisher Information. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)* (2023).
- [18] Andreas Kirsch and Yarin Gal. 2022. Unifying approaches in active learning and active sampling via fisher information and information-theoretic quantities. *Transactions on Machine Learning Research (TMLR)* (2022).
- [19] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. 2022. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)* (2022).
- [20] Chih-Chin Lai and Cheng-Chih Tsai. 2010. Digital Image Watermarking Using Discrete Wavelet Transform and Singular Value Decomposition. *IEEE Transactions on Instrumentation and Measurement* (2010).
- [21] Chenxin Li, Brandon Y Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang. 2023. StegaNeRF: Embedding Invisible Information within Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [22] Ruiqi Li and Yiu-ming Cheung. 2024. Variational multi-scale representation for estimating uncertainty in 3d gaussian splatting. *Advances in Neural Information Processing Systems* 37 (2024), 87934–87958.
- [23] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. 2023. CopyNeRF: Protecting the CopyRight of Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [24] Ziyuan Luo, Jun Liu, Haoliang Li, Anderson Rocha, and Renjie Wan. 2025. MantleMark: Migrating Watermarks from Multi-View Images to Radiance Fields via Frequency Modulation. *Authorea Preprints* (2025).
- [25] Ziyuan Luo, Anderson Rocha, Boxin Shi, Qing Guo, Haoliang Li, and Renjie Wan. 2025. The nerf signature: Codebook-aided watermarking for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).
- [26] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. 2022. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1532–1542.
- [27] Gal Metzger, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [28] Zhangkai Ni, Peiqi Yang, Wenhan Yang, Hanli Wang, Lin Ma, and Sam Kwong. 2024. ColNeRF: Collaboration for Generalizable Sparse Input Neural Radiance Field. In *The Conference on Artificial Intelligence (AAAI)*.
- [29] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [30] Thottempudi Pardhu and Bhaskara Rao Perli. 2016. Digital image watermarking in frequency domain. In *Proceedings of International Conference on Communication and Signal Processing (ICCSPP)*.
- [31] Emil Praun, Hugues Hoppe, and Adam Finkelstein. 1999. Robust mesh watermarking. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques (PACMCGIT)*.
- [32] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. 2025. Watermark Anything with Localized Messages. In *International Conference on Learning Representations (ICLR)*.
- [33] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2023. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023).
- [34] Jeongho Son, Dongkyu Kim, Hak-Yeol Choi, Han-Ul Jang, and Sunghee Choi. 2017. Perceptual 3D Watermarking Using Mesh Saliency. In *Proceedings of International Conference on Information Science and Applications (ICISA)*.
- [35] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. 2024. Geometry cloak: Preventing tgs-based 3d reconstruction from copyrighted images. *Advances in Neural Information Processing Systems* 37 (2024), 119361–119385.
- [36] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. 2024. Protecting NeRFs' Copyright via Plug-And-Play Watermarking Base Model. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [37] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. 2024. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [38] Matthew Tancik, Ben Mildenhall, and Ren Ng. 2020. StegaStamp: Invisible Hyperlinks in Physical Photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [39] Jiayang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2024. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation. In *European Conference on Computer Vision (ECCV)*.
- [40] Jiayang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2024. Dream-gaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations (ICLR)*.
- [41] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. 2023. Sparf: Neural radiance fields from sparse and noisy poses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [42] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. 1994. A digital watermark. In *Proceedings of International Conference on Image Processing (ICIP)*.
- [43] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems NeurIPS* (2017).
- [44] Peng Wang and Yichun Shi. 2023. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201* (2023).
- [45] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srivivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas

- Funkhouser. 2021. IBRNet: Learning Multi-View Image-Based Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [46] Ruofei Wang, Renjie Wan, Zongyu Guo, Qing Guo, and Rui Huang. 2024. Spy-Watermark: Robust Invisible Watermarking for Backdoor Attack. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- [47] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. 2019. High-Capacity Convolutional Video Steganography with Temporal Residual Modeling. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*.
- [48] Eric Wengrowski and Kristin Dana. 2019. Light Field Messaging with Deep Photographic Steganography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [49] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [50] Simon See Renjie Wan Xiufeng Huang, Ka Chun Cheung. 2024. GeometrySticker: Enabling Ownership Claim of Recolorized Neural Radiance Fields. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [51] Yiu-ming Cheung Ka Chun Cheung Simon See Renjie Wan Xiufeng Huang, Ruiqi Li. 2024. GaussianMarker: Uncertainty-Aware Copyright Protection of 3D Gaussian Splatting. *Neural Information Processing Systems (NeurIPS) (2024)*.
- [52] Yinghao Xu, Zifan Shi, Wang Yifan, Sida Peng, Ceyuan Yang, Yujun Shen, and Wetzstein Gordon. 2024. GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation. *European Conference on Computer Vision (ECCV) (2024)*.
- [53] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. 2024. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *International Conference on Learning Representations (ICLR)*.
- [54] Peng Yang, Yingjie Lao, and Ping Li. 2021. Robust watermarking for deep neural networks via bi-level optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- [55] Innfarn Yoo, Huiwen Chang, Xiyang Luo, Ondrej Stava, Ce Liu, Peyman Milanfar, and Feng Yang. 2022. 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)*.
- [56] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [57] Chong Yu. 2020. Attention based data hiding with generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1120–1128.
- [58] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. 2020. UDH: Universal deep hiding for steganography, watermarking, and light field messaging. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- [59] Honglei Zhang, Hu Wang, Yuanzhouhan Cao, Chunhua Shen, and Yidong Li. 2020. Robust data hiding using inverse gradient attention. *arXiv preprint arXiv:2011.10850 (2020)*.
- [60] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. 2024. GS-LRM: Large Reconstruction Model for 3D Gaussian Splatting. *European Conference on Computer Vision (ECCV) (2024)*.
- [61] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285 (2019)*.
- [62] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [63] Han Fang Zhaoyang Jia and Weiming Zhang. 2021. MBRS: Enhancing Robustness of DNN-based Watermarking by Mini-Batch of Real and Simulated JPEG Compression. In *ACM MULTIMEDIA (MM)*.
- [64] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. HiDDeN: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- [65] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. 2024. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.