X-SG^{2}S: Safe and Generalizable Gaussian Splatting with X-dimensional Watermarks

Zihang Cheng[†1], Wentao Bao[†2], Huiping Zhuang[⋆1], Chun Li[3], Xin Meng[4], Ziqian Zeng[1], Cen Chen[1], Ming Li[⋆5], and F. Richard Yu[6]

[1] South China University of Technology, Guangzhou Guangdong 511436, China
[2] Independent Researcher, USA
[3] Shenzhen MSU–BIT University, Shenzhen Guangdong 518172, China
[4] Peking University, Haidian District Beijing 100871, China
[5] School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen Guangdong 518172, China
[6] Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada
ftcldj@mail.scut.edu.cn ming.li@u.nus.edu

Abstract. 3D Gaussian Splatting (3DGS) has been widely used in 3D reconstruction and 3D generation. However, the rapid adoption of 3D Gaussian Splatting raises growing concerns about information leakage and unauthorized use, urging the exploration of effective watermarking techniques. However, existing methods are limited by low capacity, fragility under geometric perturbations, and the infeasible requirement for costly fine-tuning or pipeline modifications, motivating the need for a generalizable, feed-forward framework capable of robust multi-modal embedding with minimal intrusion. In this paper, we propose a new framework X-SG^{2}S which can simultaneously inject 1D to 3D watermarks for copyright protection, while keeping the high fidelity of original 3DGS scenes. Specifically, we first split the watermarks into message patches. A self-adaptive gate is developed to select the injection positions of the watermark messages. Then, we use an XD (multi-dimensional) injection head to inject multi-modal messages into sorted 3DGS points. To restore watermarking messages, a learnable gate is developed to recognize the watermarked locations, from which our XD-extraction heads are used to restore hidden messages. X-SG^{2}S is the first framework to unify 1D-to-3D watermarking and enable simultaneous multi-modal watermark embedding in 3DGS, achieving this with minimal rendering interference and zero modifications to parameters or pipelines. Extensive experiments demonstrate that X-SG^{2}S effectively preserves consistency between the watermark and the original 3DGS, exhibits robustness against model degradation, and maintains accurate judgment capabilities.

Keywords: Copyright Preservation · Multi Modal Watermarks · 3DGS

1 Introduction

With the growing demand for photorealistic 3D content, developing efficient and generalizable 3D representations has become a central research goal. Across do-

† Equal contribution
⋆ Corresponding authors

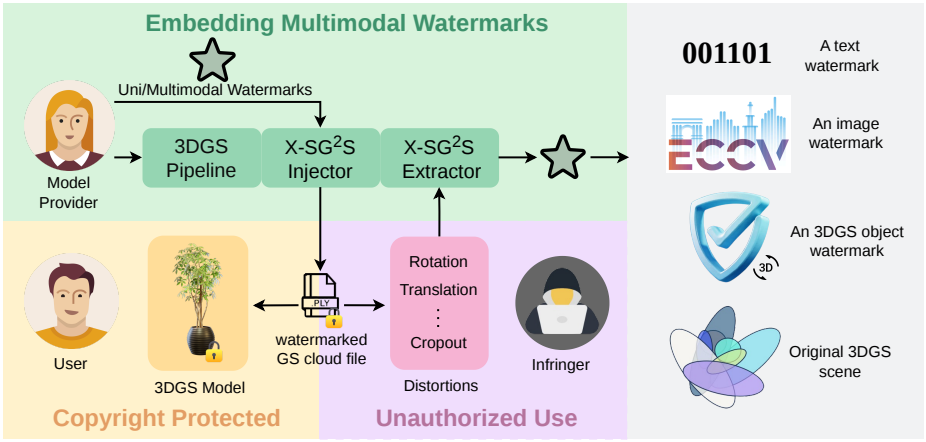

Fig. 1: Application Scenarios: After training, a 3DGS model can be paired with an X-SG^{2}S injector to embed multimodal watermarks e.g., text ("001101"), images (ECCV logo), or small 3D objects for branding. Outputs can later be verified via the extractor to confirm model usage. 2) The watermarks can be detectable even under various of hostile attacks.

mains such as healthcare, engineering, cultural heritage, VR/AR, and digital media, these techniques bridge the physical and digital worlds. Recent advances in 3DGS demonstrate its strong potential as an efficient and expressive representation for high-quality, real-time 3D reconstruction and rendering.

However, the rapid adoption of 3D Gaussian Splatting raises growing concerns about information leakage and unauthorized use which urgent to explore effective watermarking techniques. In practice, creators upload 3DGS assets to website platforms, where infringers can download, modify, and re-upload them, thereby violating the original creator's rights. Thus, platforms require an invisible yet verifiable watermark that embeds rich multimodal metadata—such as creator ID or copyright protocol bits, QR codes, or 3D object brands logo. With these watermarks, platforms can be immediately notified when copyright is infringed. However, prior works only embed *error-prone* and *extremely low-capacity* bit watermarks into 3DGS and suffer degradation under geometric perturbations, posing security risks. Moreover, all these methods require fine-tuning the original model for watermark embedding, which is infeasible without multi-view images of the scene and demands significant computational cost and time [15,6,16]. Meanwhile, some watermarking methods alter the generation pipeline or the parameter structure of 3DGS, thereby limiting their generality [37,21]. A generalizable feed-forward multimodal watermarking framework can address these issues by enabling richer semantic embedding, and more robust, general verification with minimal intrusion to the 3DGS pipeline.

To bridge this gap, we propose X-SG^{2}S, a simple, effective, and flexible watermarking framework for 3D Gaussian Splatting. It is a feed-forward network,

directly adding multimodal watermarks to partial spherical harmonic parameters and replacing the original which can avoid finetuning the model by using its training datasets. Meanwhile, our patch-wise and selective injection mechanism effectively preserves scene fidelity while isolating different modalities. Specifically, we first divide each type of watermarks into patches and introduce modality-specific redundancy to improve robustness during extraction. To enhance the robustness of image watermark recovery under partial corruption, we introduce a feature-sparse DCAE [4] to reconstruct watermarked images from incomplete or noisy features. Next, we design a self-adaptive gate that learns to select optimal insertion points by modeling the interaction between the watermark content and the 3DGS representation. An X-dimensional (XD) injection head is used to embed each modality independently, while a learnable gate identifies and records the embedding positions. Finally, the embedded messages are retrieved using the corresponding XD extraction heads.

X-SG^{2}S enables simultaneous embedding of 3D models, images, or binary messages into an existing 3DGS scene and supports accurate message extraction through a lightweight and non-intrusive process. Besides, it requires no fine-tuning or redesign of 3DGS pipeline, making it compatible with both feed-forward architectures and iterative generative models such as diffusion models. Extensive experiments demonstrate that X-SG^{2}S can effectively embed multimodal watermarks without degrading rendering quality, while offering high fidelity, robustness, reliability, and flexibility. In a nutshell, the contributions and advantages of our X-SG^{2}S are summarized as follows:

- We propose **X-SG^{2}S**, a unified and feed-forward watermarking framework for 3DGS that completes watermark embedding within seconds.
- X-SG^{2}S exhibits large capacity and strong versatility, that supports the simultaneous embedding of **1D** binary messages, **2D** images, and **3D** objects into a single 3DGS scene.
- Our patchifying strategy and X-SG^{2}S backbone demonstrate robust security and high fidelity, preserving watermark-decoding quality even under a wide range of attacks.

2 Related Work

2.1 Generalizable GS

3D Gaussian Splatting [18] has emerged as an efficient representation for high-quality 3D reconstruction, leveraging millions of anisotropic Gaussians for continuous and differentiable rendering. Building on this foundation, subsequent works have extended 3DGS to both object- and scene-level reconstruction. Methods such as Splatter Image [33] and SF3D [2] focus on compact or category-specific objects, while MV-Splat [5] and PixelSplat [3] improve scalability and multi-view consistency for complex scenes. Beyond reconstruction, diffusion-based approaches like DiffGS [40] further integrate generative modeling, enabling image-to-3D, text-to-3D and point cloud to 3DGS generation. Collectively, these

studies demonstrate the expanding versatility of 3DGS across reconstruction and generation tasks.

2.2 Watermarking in Deep Learning

Digital watermarking embeds imperceptible signals to verify copyright and protect multimedia ownership. Early image watermarking methods such as HiDDeN [41] and HiNet [17] use CNNs to encode binary messages or watermark images while preserving visual quality, while VideoSeal [9] leverages temporal coherence for robust video watermarking. More recently, diffusion-based approaches like Videomark [14] and Safe-SD [26] inject ownership signals into the stochastic noise trajectories of generative diffusion processes. These methods highlight watermarking's dual role in safeguarding intellectual property and ensuring content traceability through imperceptible yet resilient signal embedding.

2.3 3D Steganography and watermarking

In 3D generative era [23,22,8,38,39,13,12], some researches have made 3D steganography achieve good results in explicit geometry like meshes and point clouds [29,42,10] or implicit geometer like NeRF [20,25]. 3DGS is a new technology to represent explicit geometry. However, few works have been researched in steganography for 3DGS. GS-Hider [37] design a coupled secured feature attribute to replace the original 3DGS's spherical harmonics coefficients and then use a scene decoder and a message decoder to disentangle the original RGB scene and the hidden message. Splat in Splat [11] first train two scene by using the same location and shape of the cover scene and different color and opacity. Then they inject the secret scene into the cover one by an unlearnable method and use an AE to map the transparency variation of each point. This method can work only when you have the entire multi view datasets. And for every scene, you should learn a new AE which is very fussy. GaussianStego [21] uses DINO to add image watermarks and use U-Net to extract them via multi rendered views. GaussianMarker [15] can only embed 48 bit watermarks. The added information might be insufficient in specific contexts.

3 Method

3.1 Proposed Method

Our goal of 3D Gaussian Splatting watermarking is to inject multimodal watermarks into a 3DGS scene without compromising structure or rendering quality. The following objectives are desired in this task:

- **3DGS Steganography:** embed auxiliary information into 3DGS scenes without damaging visual and structural fidelity.
- **Copyright protection:** embed user-defined watermarks for ownership verification, enabling detection of unauthorized usage by recovering injected watermarks.

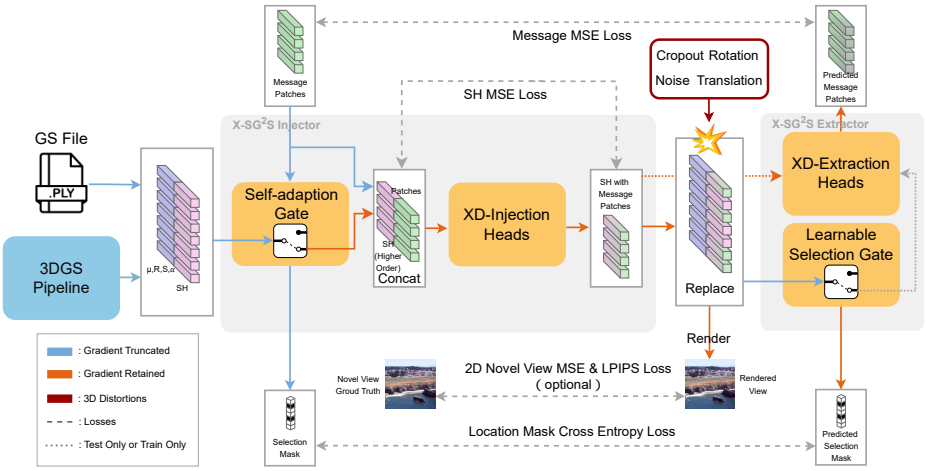

Fig. 2: **Overview of X-SG^{2}S.** We first use patchify strategy splitting the watermark into small patches. At injection stage, adaptive selection gate picks the 3DGS locations best suited for embedding given patches. XD-injection head embeds the patches into the high-order spherical-harmonic coefficients of the selected Gaussians. At extraction stage, a recognition gate retrieves the watermarked Gaussians and XD-extraction head decodes those patches. We truncate the gradient before the learnable selection gate and feed only the watermarked GS points under various 3D distortions to the XD-extraction head while training.

- **Multimodal embedding:** support text, image, and 3D object watermarks within one scene, enhancing the richness of the watermark while ensuring modality independence and accurate retrieval.
- **General and non-intrusive:** preserves the original 3DGS parameters and ensures seamless integration without any pipeline architectural modifications.
- **Fine-tuning free:** no fine-tuning and no access to the images originally used to train the 3DGS model.

3.2 Preliminary

3D Gaussian Splatting [18] is a rasterization-based representation enabling real-time, differentiable rendering of 3D scenes. A scene is represented as a set of anisotropic 3D Gaussians that collectively approximate both geometry and appearance.

Gaussian Representation. Each Gaussian $\mathcal{G}_i$ is defined by its center $\boldsymbol{\mu}_i \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ (parameterized by a scale vector $\boldsymbol{s}_i$ and a rotation quaternion $\boldsymbol{q}_i$), opacity $\alpha_i \in [0,1]$, and view-dependent color $\boldsymbol{c}_i(\boldsymbol{v})$ represented via

Spherical Harmonics (SH) of order k. Its spatial density is:

$$\rho_i(\boldsymbol{x}) = \exp\left[-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right] \tag{1}$$

A complete scene is thus $\mathcal{G} = \{(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \boldsymbol{c}_i)\}_{i=1}^N$.

Rasterization and Rendering. Given a camera with projection matrix P and world-to-camera transform W, each Gaussian is projected to the image plane as

$$\hat{\boldsymbol{\mu}}_i = PW\boldsymbol{\mu}_i \ , \quad \hat{\boldsymbol{\Sigma}}_i = J_i \boldsymbol{\Sigma}_i J_i^\top \tag{2}$$

where J_i is the Jacobian of the local projection at $\boldsymbol{\mu}_i$. The per-pixel contribution is

$$\sigma_i(\mathbf{p}) = \alpha_i \exp\left[-\tfrac{1}{2}(\mathbf{p}-\hat{\boldsymbol{\mu}}_i)^\top \hat{\boldsymbol{\Sigma}}_i^{-1}(\mathbf{p}-\hat{\boldsymbol{\mu}}_i)\right] \tag{3}$$

Final colors are accumulated via alpha compositing along the viewing ray:

$$\mathbf{C}[\mathbf{p}] = \sum_{i=1}^{N} \boldsymbol{c}_i(\boldsymbol{v})\, \sigma_i(\mathbf{p}) \prod_{j<i}(1-\sigma_j(\mathbf{p})) \tag{4}$$

This formulation enables efficient, differentiable, and photorealistic rendering of complex 3D scenes in real time.

3.3 Methodology Overview

Challenges. 3DGS represents scenes as unordered point sets with rich parameters, posing unique challenges for watermarking: (1) Ordering invariance: the lack of inherent point order complicates the selection of points for watermarking, often necessitating global modifications [29] that waste capacity or degrade quality. (2) Localization ambiguity: embedding in subsets of points lacks reliable indexing, especially under multiple watermarks. (3) Limited per-point capacity: each Gaussian stores minimal information; efficiently encoding large payloads requires fragmentation and precise reconstruction strategies. (4) Robustness issue: when certain points are missing or corrupted, watermark recovery must rely solely on the remaining or perturbed fragments which is usually challenging.

Solution. To address these challenges, we develop X-SG^{2}S for 3DGS watermarking as depicted in Figure 2. We propose a modality-specific watermark splitting strategy. These patches are then processed by four modules that jointly inject and faithfully retrieve the watermark while keeping the original scene minimally perturbed: (1) an adaptive self-adaption gate for adaptively selecting GS points to use. (2) a multi-head MLP-based message injector for integrating information into SH parameters. (3) a learnable selection gate for recognizing the location of injected GS points. (4) a multi-head MLP-based message extractor for restoring message from the injected GS points. X-SG^{2}S injector is made up of (1) and (2), extractor is made up of (3) and (4).

During training, the self-adaption gate generates a mask indicating the watermark injection locations and let the XD-injection head add the watermark patches. The learnable selection gate identifies those positions from the watermarked Gaussian parameters under various of 3D distortions and performs learning under the guidance of the mask. Simultaneously, the XD-extraction head extracts the patches from the Gaussian parameters watermarked by XD-injection head. During inference, only the Gaussian points selected by the learnable selection gate are sent to the XD-extraction head for watermark decoding.

3.4 XD Watermark Representation

To solve the challenges (3) and (4) introduced in the overview, watermarks are split into small patches with modality-aware redundancy. Patchfying reduces the information capacity per patch, avoiding overloading a single Gaussian point, which would not lead to information loss during recovery. Redundancy expansion further allows the watermarks to be reconstructed from their surviving counterparts when partial crops occur, yielding a robustness boost.

Patchifying Strategy. We handle different modalities with modality-specific patching strategies. For 1D and 3D data, which are inherently discrete, we simply duplicate the original data multiple times to create redundancy. This allows accurate watermark recovery from any randomly sampled subset, ensuring robustness without additional processing. Features encoded by DCAE are split to patches and augmented with (i) position encoding and (ii) a compact summary of its neighboring patches, so that every token knows "where it is" in the feature map and the missing ones can be reconstructed from the survivors when partial corruptions occur. In the Figure 3, we show our **feature-sparse DCAE**. DCAE [4] encodes an image of resolution n into a feature map of $\frac{n}{64} \times \frac{n}{64} \times d$, where the feature dimension is d. Then, the feature maps are flattened as a sequence with the shape of $\frac{n^2}{64^2} \times d$. Four up-sampling convolution layers are used to create redundant context for this sequence, with fixed position encoding added. To simulate a hostile attack, we randomly remove some patches. Benefited by the context sequence, our module is capable of restoring the whole sequence by any subset. Specifically, feature-sparse DCAE uses an MLP to predict the fixed position encoding. Then, it performs the nearest-neighbor search by comparing the predicted encoding and the complete encoding, which determines the patch locations and let all the patches be placed in the correct position. Then, down-sampling convolution layers and a refinement MLP are used to restore the original feature sequence. Finally, we rearrange the feature sequence and decode the watermark. More details about feature-sparse DCAE are introduced in the Appendix.

Message Embedding. To enable uniform interaction, we use separate linear layers to map 1D, 2D, and 3D messages into a higher dimension latent space which provides rich information for computing watermarking locations.

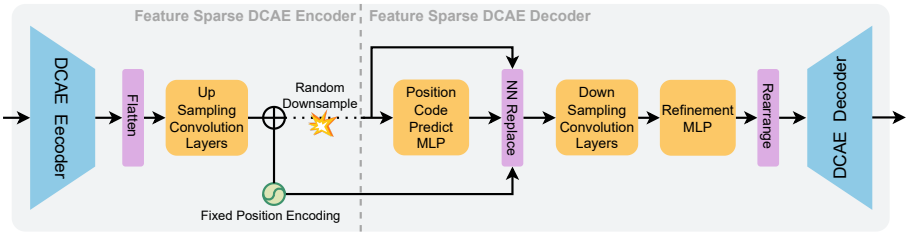

Fig. 3: The structure of the feature-sparse DCAE.

3.5 Self-Adaptive Selection of Injection Position

Selecting optimal 3DGS points for message injection while preserving scene fidelity is non-trivial in practice. To tackle this challenge, we consider the point selection from two perspectives: the "self-score" and the "cross-score". The "self-score" measures how much a point contributes to the overall 3DGS appearance. The "cross-score" reflects the alignment between the embedded message and the point cloud structure. Eventually, a top-k mask from the product of the two scores can be used to determine where to inject watermarks. In light of the two scores, we develop a self-adaptation gate, as illustrated in Figure 4, which will be entailed below.

Self-score. Computing the self-score requires each point to interact with the entire 3DGS. But vanilla Transformer incurs a prohibitive memory cost. To reduce memory complexity while retaining the interaction capability, we adopt Induced Set Attention [19], which is efficient and order-invariant. Given n Gaussian point embeddings $X \in \mathbb{R}^{n\times d}$ and m learnable inducing tokens $I \in \mathbb{R}^{m\times d}$, we compute

$$\begin{aligned} I^* &= f(I, X) \\ O_s &= f(X, I^*) \\ \text{self-score} &= \sigma\big(h(O_s)\big) \end{aligned} \tag{5}$$

where $f(\cdot,\cdot)$ is multi head attention block and $h(\cdot)$ is a linear layer. We use $m = 1$ and one attention head, reducing complexity from $O(n^2)$ to $O(n)$.

Cross-score. To compute the "cross-score", we use Efficient Attention [32]. Efficient attention (EA) not only drastically cuts computation but also preserves rich interaction between the 3DGS and watermarks. Given l message patches $Y \in \mathbb{R}^{l\times d}$, the attention output is

$$\begin{aligned} O_c &= \sigma_{\text{row}}(X)\left[\sigma_{\text{col}}(Y)^\top Y\right] \\ \text{cross-score} &= \sigma(h(O_c)) \end{aligned} \tag{6}$$

Finally, the gate score is obtained by multiplying the "self-score" and "cross-score" element-wise. Table 6 shows the efficiency of our self-adaptive gate.

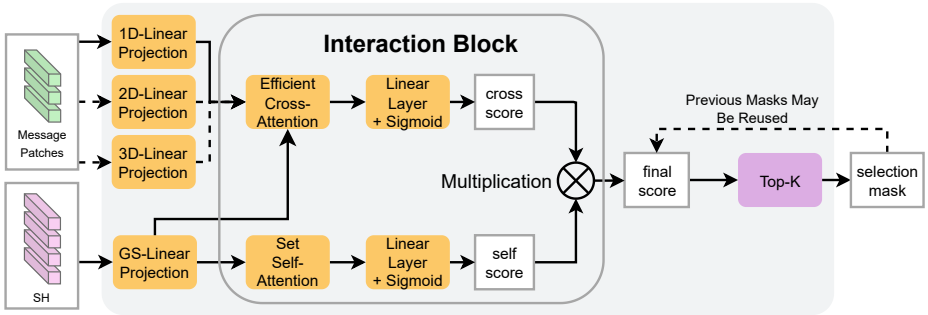

Fig. 4: The structure of the self-adaptation gate.

3.6 Learnable Selection Gate

The previous self-adaptation gate can produce a mask when injecting a watermark into a 3DGS scene. However, another equally important task is to recover the watermark from the injected positions, which has to be learned by a module. We treat the predicted masks from the self-adaptation gate as the ground truth. The learnable selection gate regards the point set with added watermarks as input to find out the locations where the information is added. The gate is a four-layer MLP to predict the true location. During training, we detach the gradients before the learnable selection gate, ensuring it only learns where to embed the watermark.

3.7 XD-Injection/Extraction Heads

XD-Injection modules embed and extraction heads extract watermark patches. For each modality, we use a separate MLP following PointNet [30], typically with 3–5 layers. We select GS points from target locations suggested by the self-adaptive gate and extract their high-order spherical harmonic (SH) parameters from three channels. These are concatenated with the message patches and passed through a multi-head injector to produce SH parameters carrying the embedded information. And for the extractor, it takes the SH parameters of injected points found out by the learnable selection gate as input and extracts the messages from them. The extracted messages will be optimal by the prediction loss as well.

3.8 Training Loss

To effectively train X-SG^{2}S, we develop a comprehensive loss function comprising four components: message patches loss, location mask loss, SH parameters loss, and other optional perceptual losses.

$$\begin{aligned} LOSS = \phi\mathcal{L}_{\text{1D-BCE}} + \theta\mathcal{L}_{\text{2D-MSE}} + \delta\mathcal{L}_{\text{3D-MSE}} \\ +\mathcal{L}_{\text{Mask-BCE}} + \gamma\mathcal{L}_{\text{SH-MSE}} + \mathcal{L}_{\text{Optional}} \end{aligned} \tag{7}$$

The message patches loss enforces accurate recovery of the injected information by ensuring consistency between the embedded and extracted messages. This is fulfilled by a cross-entropy loss $\mathcal{L}_{\text{1D-BCE}}$ for 1D messages and mean squared error (MSE) losses $\mathcal{L}_{\text{2D-MSE}}$ and $\mathcal{L}_{\text{3D-MSE}}$ for 2D and 3D messages, respectively. Mask cross-entropy loss $\mathcal{L}_{\text{Mask-BCE}}$ guides the learnable selection gate to precisely identify regions for message embedding. The SH parameters loss penalizes large deviations in spherical harmonic (SH) coefficients through an MSE loss $\mathcal{L}_{\text{SH-MSE}}$. Additionally, other losses, such as 2D view reconstruction loss and LPIPS loss, can be incorporated when leveraging pretrained 3DGS pipelines to further enhance the preservation of SH parameters and overall rendering fidelity.

4 Experiments

4.1 Experimental Setup

Datasets. We use a pretrained model MVSplat for real time inference. We use large-scale ACID [24] datasets to let MVSplat [5] generate the 3DGS scenes. For 1D dataset, we randomly generate a sequence of binary message. For each bit, it has the same probability to be 1 or 0. For 2D dataset, we use Logo-2K [34] dataset. For 3D dataset, we first download a subset of Objaverse [7], and use Gamba [31] to generate the 3DGS objects. They are then saved in .ply format as a 3D dataset.

Specifically, ACID contains nature scenes captured by aerial drones, which are split into 11,075 training scenes and 1,972 testing scenes. After reconstructing Logo-2K dataset, it has 110313 training logos and 28415 testing logos. 3D dataset made by objaverse and Gamba contains 353 training files and 94 testing files. To ensure the number of datasets is aligned, we reuse the GS cloud files while training.

Metrics. For original GS scenes and GS objects, we use pixel-level PSNR, patch-level SSIM [35], and feature-level LPIPS [36] to measure the completeness of the original scenes and the quality of restoration of the GS objects. For a fair comparison, all the scenes are rendered to 256×256 resolutions while all the objects are rendered to 512×512 resolutions. For 1D data, we use precision rate to measure the extraction accuracy of binary data. For 2D data, we use pixel-level PSNR to quantization the restoration of the 2D images.

4.2 Implementation Details

X-SG^{2}S is implemented with PyTorch, along with an off-the-shelf 3DGS render implemented in CUDA. All models are trained on a single RTX A6000ada with the Adam optimizer. The optional loss functions include 2D rendered image MSE loss and LPIPS loss, with weights of 1 and 0.05, respectively. And other hyperparameter are $\gamma = 1$, $\phi = 0.005$, $\theta = 2$, $\delta = 1.5$. We choose a binary message of 48bit×1024 as a 1D watermark and a 3DGS object with 10,000 points as a 3D watermark. For the 2D watermark, we use feature sparse DCAE to get 16×2048 message patches. The number of higher-order SH parameters is set to 48 for each GS point.

Fig. 5: This image illustrates the different effects of watermarking all SH coefficients versus watermarking only the high-order SH coefficients. The odd columns show X-SG^{2}S injects the watermark to all the SH while the even columns show the watermarks are only injected to the higher order SH.

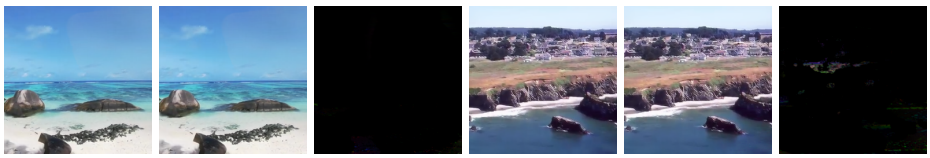

Fig. 6: This image demonstrates the effectiveness of our approach. The first column shows the original scene images rendered by MVSplat. The second column displays the images after watermark embedding. The third column presents the 10×residuals between the two images. It can be seen that our model is capable of effectively concealing the watermark.

4.3 Fidelity

First, we evaluate where to embed the watermark patches within the spherical harmonic (SH) coefficients of 3DGS can minimize the distortion to the original scene. To this end, we conduct an experiment by injecting identical watermarks into all SH coefficients and only into the higher-order SH coefficients separately. According to figure 5, we show the importance of SH. It seems that only inject watermarks to higher order SH can achieve better results. At the same time, when only adding watermarks to higher order SH, the watermarks cannot be detected by the naked eye from the visual perspective which means we protect the original scene well and hide the watermarks well. Meanwhile,we have drawn the residual map of the invisible watermark in figure 6. It can be observed that our method can effectively hide the watermark without detection. In figure 7, we also shows 2D and 3D watermark extracted by our X-SG^{2}S. The decoded watermarks are clearly legible and of high quality which means our X-SG^{2}S can inject and extract watermarks well.

4.4 Robustness

We have subjected the Gaussians to 3D degradation using random pruning, rotation, translation adding gaussian noise methods. The following distortions for message extraction are considered: Gaussian Noise($\sigma = 0.1$), Rotation(random select between $+\pi/6$ and $-\pi/6$), Translation(random select between $[0, 1000]^3$),

Table 1: Comparison of the quantitative performance of the original scenes and watermarks under various perturbations on ACID [24] and Mip-NeRF360 [1] dataset.

Attack Type	**Model Ability**				
	Acc	PSNR (image)	PSNR (3D obj)	SSIM (3D obj)	LPIPS↓(3D obj)
None	100	22.791	23.017	0.836	0.138
Noise to SH($\sigma = 0.05$)	96.22	20.008	20.307	0.796	0.219
Translation($t = [0, 1000]^3$)	99.31	22.669	23.010	0.835	0.139
Rotation($r = \pm\pi/6$)	98.56	22.139	22.718	0.834	0.141
Cropout($cr = 0.1$)	100	22.539	23.005	0.835	0.139

Table 2: Comparison of bit accuracy under various perturbations on Blender [28] and LLFF [27] datasets. We also report the watermarking speed and watermarking capability. We show the results on 48-bit messages. Bold indicate the best performance.

Method	**Model Ability**		**Bit Acc** ↑ (%)				
	MM wms	wm spd	None	Noise to SH ($\sigma = 0.05$)	Translation ($t = [0, 10^3]^3$)	Rotation ($r = \pm\pi/6$)	Cropout ($cr = 0.1$)
3DGS [18] w/ fine-tuning	✗	3-5 mins	69.79	69.70	68.78	65.88	64.84
GaussianMarker [15] 2D dec	✗	3-5 mins	97.85	57.05	59.07	53.88	48.23
GaussianMarker [15] 3D dec	✗	3-5 mins	**100**	**99.90**	98.95	95.83	92.70
GuardSplat [6]	✗	3-5 mins	90.98	67.10	67.77	66.13	71.32
3D-GSW [16]	✗	3-5 mins	93.72	84.54	81.67	82.33	91.87
Ours	✓	**3 secs**	**100**	94.12	**99.07**	**96.56**	**95.01**

Cropout(10% of the original). Quantitative metrics are shown in Table 1. The results indicate that our method effectively withstands the degradation process.

4.5 Compare with Other Methods

We compare X-SG^{2}S with 5 baselines on binary watermark to guarantee a fair comparison: (1) 3DGS [18] with fine-tuning, (2) GaussianMarker 2D Decoder [15], (3) GaussianMarker 3D Decoder [15], (4) GuardSplat [6] and (5) 3D-GSW [16]. We embed 48-bit watermarks into the same scenes using these different methods. Table 2 demonstrate that our approach effectively withstands external 3D perturbations and achieves state-of-the-art performance.

4.6 Precise Identification

In the real case, we cannot indiscriminately detect watermarks. That is to say, we cannot identify a watermark in a model that does not have one. To illustrate this, we randomly select some scenes without watermarks and observe that our model fails to detect any watermark. Table 3 demonstrates that our XD extraction head can not extract watermark amount unwatermarked 3DGS model which can effectively reduce false positives without the need for an additional discriminator.

Fig. 7: The image shows watermarks extracted from 3DGS scenes by X-SG^{2}S. The first column shows the original image watermarks provide by Logo-2K [34]. The second column displays the image watermarks extracted by X-SG^{2}S. The third column displays the GS objects provided by Gamba [31]. The final column presents the GS objects extracted by X-SG^{2}S.

Table 3: To verify whether the detection model can distinguish if the original model contains a watermark. Results show X-SG^{2}S can unambiguously determine.

	GS w/o wm	GS w wm
PSNR (org)	27.826	27.804
SSIM (org)	0.871	0.871
LPIPS (org)↓	0.123	0.126
Acc	0.395	1.000
PSNR (image)	7.719	22.791
PSNR (3D obj)	11.934	23.017
SSIM (3D obj)	0.835	0.836
LPIPS (3D obj)↓	0.319	0.138

Table 4: Single/Multi-modality watermark injection and extraction. X-SG^{2}S supports watermark injection for both single-modal and multi-modal data.

Method	1D	2D	3D	XD	Org
PSNR(org)	27.814	27.811	27.810	27.804	27.826
SSIM(org)	0.871	0.871	0.871	0.871	0.871
LPIPS(org)↓	0.124	0.125	0.125	0.126	0.123
Acc	1.000	/	/	1.000	1.000
PSNR(image)	/	22.804	/	22.791	29.755
PSNR(3D obj)	/	/	23.044	23.017	28.807
SSIM(3D obj)	/	/	0.840	0.836	0.942
LPIPS(3D obj)↓	/	/	0.137	0.138	0.067

4.7 Ablation Results

The Effectiveness Of Our Interaction Block. We design three different structures of interaction block and conduct ablation experiments respectively: (1) "Interactive": enabling interactions between Gaussian points and watermark blocks as shown in Figure 4. (2) "Normal": using 3DGS original points as input and passing them through n layers of MLP gate. (3) "Random": randomly selecting k locations to add an extra message that can not be learned. Table 6 demonstrates that enabling interactions between points and watermarks significantly improves performance.

Single Modality Injection Test. In Table 4, we test our X-SG^{2}S on adding sigle watermark. Because the different modalities are mutually non-interfering during embedding, our model can flexibly inject either single-modal or multi-modal watermarks while guaranteeing their exact recovery. We can see that X-SG^{2}S can well adapt the task of 1 to 3D watermarking and the performance of watermark extraction are almost identical.

The Exploration of Loss Function. Here we set the weights of optional losses to 0. From Table 5, we can see that this way is feasible. However, X-SG^{2}S trained

Table 5: Comparison with and without optional loss functions. Results show X-SG^{2}S trained with optional loss performs better.

Method	w/o optional losses	w optional losses
PSNR (org)	27.757	**27.804**
SSIM (org)	0.869	**0.871**
LPIPS (org)↓	0.128	**0.126**
Acc	0.979	**1.000**
PSNR (image)	21.923	**22.791**
PSNR (3D obj)	22.647	**23.017**
SSIM (3D obj)	0.833	**0.836**
LPIPS (3D obj)↓	0.140	**0.138**

Table 6: Comparing different structures of interaction block. Our module adaptively aligns 3DGS and watermarks to optimize embedding locations.

Method	Ours	Normal	Random	Org
PSNR (org)	**27.804**	24.276	21.534	27.826
SSIM (org)	**0.871**	0.824	0.529	0.871
LPIPS (org)↓	**0.126**	0.235	0.443	0.123
Acc	**1.000**	0.958	0.667	1.000
PSNR (image)	**22.791**	22.595	21.797	29.755
PSNR (3D obj)	**23.017**	21.322	20.699	28.807
SSIM (3D obj)	**0.836**	0.833	0.831	0.942
LPIPS (3D obj)↓	**0.138**	0.149	0.165	0.067

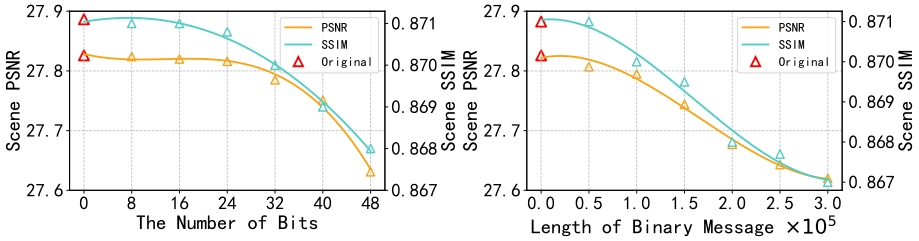

Fig. 8: The interference of watermark capacity on the original scene.

Table 7: Analysis of robustness against varying Cropout intensities. Despite increasing distortion, the watermark extraction accuracy remains stable.

Cr	Acc	PSNR(img)	PSNR(obj)	SSIM	LPIPS↓
5%	100	22.611	23.009	0.835	0.139
10%	100	22.539	23.005	0.835	0.139
15%	100	22.340	22.912	0.831	0.141
20%	98.3	22.098	22.470	0.828	0.147
25%	97.7	21.563	22.133	0.824	0.155

without optional loss performs worse. Therefore, we recommend using optional loss functions to enhance the effectiveness of the training.

Capacity vs. Distortion Trade-off. Figure 8 conducted two experiments to demonstrate how increasing watermark payload affects rendering quality: The left figure is for the number of bits (fixed at 15,000 patches), while the right is for the patch length (48 bits/patch). Both used bit-only watermarks. Results show that longer length and more patches lead to greater degradation in rendering quality but the interference remains limited.

Robustness Under Varying Perturbation Strengths. Table 7 presents the model's robustness against watermark attacks of varying intensities. To evaluate this, we subjected the model to cropout attacks with different intensity levels, which shows less than 3% accuracy drop even with 25% cropout rate.

5 Conclusion

In this paper, we introduce X-SG^{2}S, the first multimodal watermarking framework designed explicitly for 3D Gaussian Splatting. X-SG^{2}S embeds 1-, 2-, or 3-dimensional payloads into a radiance field without altering the original 3DGS parameters or requiring per-scene fine-tuning, thus eliminating time and storage overhead. By pairing a novel modality-specific patchifing method with a lightweight neural policy network, our method adaptively selects the most suitable locations in 3DGS scenes for embedding and reliably recovers the payload after severe 3D corruptions (noise, rotation, translation, cropout). Extensive experiments on diverse scenes demonstrate a higher payload capacity than existing methods while maintaining visual fidelity indistinguishable from the pristine rendering. We believe X-SG^{2}S will facilitate downstream copyright protection and authentication tasks for rapidly-growing 3DGS assets.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (62306117), the Guangdong S&T Programme (2025B0101120007), the Guangdong Basic and Applied Basic Research Foundation (2026A1515011478), and GJYC program of Guangzhou (2024D03J0005).

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
2. Boss, M., Huang, Z., Vasishta, A., Jampani, V.: Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. arXiv preprint arXiv:2408.00653 (2024)
3. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19457–19467 (2024)
4. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. arXiv preprint arXiv:2410.10733 (2024)
5. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European Conference on Computer Vision. pp. 370–386. Springer (2025)
6. Chen, Z., Wang, G., Zhu, J., Lai, J., Xie, X.: Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16325–16335 (2025)
7. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)

8. Fang, X., Easwaran, A., Genest, B., Suganthan, P.N.: Your data is not perfect: Towards cross-domain out-of-distribution detection in class-imbalanced data. Expert Systems with Applications (2025)
9. Fernandez, P., Elsahar, H., Yalniz, I.Z., Mourachko, A.: Video seal: Open and efficient video watermarking. arXiv preprint arXiv:2412.09492 (2024)
10. Ferreira, F.A., Lima, J.B.: A robust 3d point cloud watermarking method based on the graph fourier transform. Multimedia Tools and Applications **79**(3), 1921–1950 (2020)
11. Guo, Y., Huang, W., Li, Y., Li, G., Zhang, H., Hu, L., Li, J., Huang, T., Ma, L.: Splats in splats: Embedding invisible 3d watermark within gaussian splatting. arXiv preprint arXiv:2412.03121 (2024)
12. He, T., Gao, L., Song, J., Li, Y.F.: Towards open-vocabulary scene graph generation with prompt-based finetuning. In: European Conference on Computer Vision (ECCV). pp. 56–73 (2022)
13. He, T., Gao, L., Song, J., Li, Y.F.: Toward a unified transformer-based framework for scene graph generation and human-object interaction detection. IEEE Transactions on Image Processing **32**, 6274–6288 (2023)
14. Hu, X., Li, H., Li, J., Huang, Y., Liu, A.: Videomark: A distortion-free robust watermarking framework for video diffusion models. arXiv preprint arXiv:2504.16359 (2025)
15. Huang, X., Li, R., Cheung, Y.m., Cheung, K.C., See, S., Wan, R.: Gaussianmarker: Uncertainty-aware copyright protection of 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 33037–33060 (2024)
16. Jang, Y., Park, H., Yang, F., Ko, H., Choo, E., Kim, S.: 3d-gsw: 3d gaussian splatting for robust watermarking. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5938–5948 (2025)
17. Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z.: Hinet: Deep image hiding by invertible network. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4733–4742 (2021)
18. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
19. Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: International conference on machine learning. pp. 3744–3753. PMLR (2019)
20. Li, C., Feng, B.Y., Fan, Z., Pan, P., Wang, Z.: Steganerf: Embedding invisible information within neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 441–453 (2023)
21. Li, C., Liu, H., Fan, Z., Li, W., Liu, Y., Pan, P., Yuan, Y.: Gaussianstego: A generalizable stenography pipeline for generative 3d gaussians splatting. arXiv preprint arXiv:2407.01301 (2024)
22. Li, W., Su, X., Cao, Y., Xu, H., Xia, X., You, S., Chen, Y., Xu, C.: Sentinel-vla: A metacognitive vla model with active status monitoring for dynamic reasoning and error recovery (2026), https://arxiv.org/abs/2605.01191
23. Li, W., Su, X., Niu, D., Cao, Y., Xu, H., Qu, Z., Fan, L., You, S., Xu, C.: Vla-attc: Adaptive test-time compute for vla models with relative action critic model (2026), https://arxiv.org/abs/2605.01194
24. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 14438–14447 (2020), https://api.semanticscholar.org/CorpusID:229297871

25. Luo, Z., Guo, Q., Cheung, K.C., See, S., Wan, R.: Copyrnerf: Protecting the copyright of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22401–22411 (2023)
26. Ma, Z., Jia, G., Qi, B., Zhou, B.: Safe-sd: Safe and traceable stable diffusion with text prompt trigger for invisible generative watermarking. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 7113–7122 (2024)
27. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (ToG) **38**(4), 1–14 (2019)
28. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
29. Ohbuchi, .R., Mukaiyama, .A., Takahashi, .S.: A frequency-domain approach to watermarking 3d shapes. In: Computer graphics forum. vol. 21, pp. 373–382. Wiley Online Library (2002)
30. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
31. Shen, Q., Wu, Z., Yi, X., Zhou, P., Zhang, H., Yan, S., Wang, X.: Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. arXiv preprint arXiv:2403.18795 (2024)
32. Shen, Z., Zhang, M., Zhao, H., Yi, S., Li, H.: Efficient attention: Attention with linear complexities. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 3531–3539 (2021)
33. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10208–10217 (2024)
34. Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., Wang, H., Jiang, S.: Logo-2k+: A large-scale logo dataset for scalable logo classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6194–6201 (2020)
35. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
37. Zhang, X., Meng, J., Li, R., Xu, Z., Zhang, Y., Zhang, J.: Gs-hider: Hiding messages into 3d gaussian splatting. arXiv preprint arXiv:2405.15118 (2024)
38. Zheng, J., Liu, X., Liu, W., He, L., Yan, C., Mei, T.: Gait recognition in the wild with dense 3d representations and A benchmark. In: CVPR. pp. 20228–20237 (2022)
39. Zheng, J., Liu, X., Wang, S., Wang, L., Yan, C., Liu, W.: Parsing is all you need for accurate gait recognition in the wild. In: ACM MM. pp. 116–124 (2023)
40. Zhou, J., Zhang, W., Liu, Y.S.: Diffgs: Functional gaussian splatting diffusion. ArXiv **abs/2410.19657** (2024), `https://api.semanticscholar.org/CorpusID:273638402`
41. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 657–672 (2018)

42. Zhu, X., Ye, G., Luo, X., Wei, X.: Rethinking mesh watermark: Towards highly robust and adaptable deep 3d mesh watermarking. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 7784–7792 (2024)