Guard4D: Robust Watermarking for 4D Gaussian Splatting via Decoupled Decoding

Yingying Shi[1,2], **Zhong Zhou**[1,2], **Bin Zhou**[1,3*]
[1]State Key Laboratory of Virtual Reality Technology and Systems,
School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Peace-Bay Academy of Virtual Reality, Shenyang, China
{yyshi, zz, zhoubin}@buaa.edu.cn

Abstract

4D Gaussian Splatting (4DGS) enables high-quality, real-time rendering of dynamic scenes and is becoming a popular format for sharing 4D assets. This trend calls for robust copyright watermarking that can be verified from rendered videos without access to the original scene or model parameters. However, watermarking 4DGS is challenging: verification relies on short multi-view clips whose frames vary with motion, viewpoint changes, and post-processing. More importantly, decoding from rendered frames is easily dominated by scene content (objects, textures, illumination) rather than subtle watermark traces, leading to unstable evidence across frames and poor generalization across scenes. We propose Guard4D, a decoupled watermarking framework for 4DGS. Guard4D pre-trains a general-purpose message decoder under text supervision and embeds a binary message by optimizing compact spherical-harmonic offsets while keeping geometry and opacity fixed. To improve extraction from short clips and suppress semantic interference, we introduce a temporal modeling and semantic decoupling (TMSD) module that temporally aggregates watermark information and uses clean and noise-perturbed control clips generated from the non-watermarked model to reduce the influence of semantics on watermark decoding. Extensive experiments on dynamic 4DGS scenes demonstrate the effectiveness of Guard4D. Code is available at https://github.com/shisyy/Guard4D.

1 Introduction

Three-dimensional assets have become fundamental to modern digital-content creation, virtual and augmented reality, and cultural-heritage preservation [Boboc *et al.*, 2022]. As dynamic neural assets become increasingly exchanged and commercialized, robust copyright protection and ownership verification become critical. Digital watermarking offers a practical mechanism to embed imperceptible information for attribution and verification [Narendra *et al.*, 2022], and neural watermarking further extends this capability to neural rendering models [Boenisch, 2021; Chen *et al.*, 2023]. Prior works have explored watermarking for neural representations and Gaussian splatting, mostly under static assumptions. NeRF-based approaches embed messages into radiance-field parameters but often incur expensive rendering costs [Boenisch, 2021; Chen *et al.*, 2023]. 3D Gaussian Splatting (3DGS) represents a scene with anisotropic Gaussians and spherical-harmonic (SH) appearance, supporting real-time rendering [Kerbl *et al.*, 2023], and recent methods watermark 3DGS by perturbing a subset of Gaussian parameters or optimizing lightweight residual offsets [Jang *et al.*, 2025; Chen *et al.*, 2025]. However, these designs implicitly rely on time-invariant geometry and appearance. In contrast, 4DGS models dynamics via temporally deformable Gaussians, where both geometry and view-dependent appearance evolve continuously over time [Wu *et al.*, 2024]. This shift exposes an emerging protection gap: watermarking techniques tailored to static scenes are not designed for spatiotemporally deformable Gaussian fields.

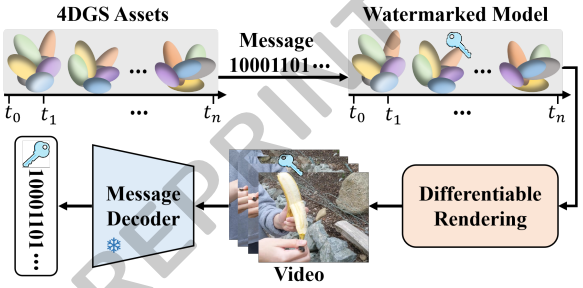

Figure 1: Given a 4DGS asset, we embed a binary message into its representation to obtain a watermarked model. The watermarked model is differentiably rendered into video, from which the private decoder extracts the embedded watermark.

Watermarking 4DGS is challenging for two reasons. First, in practice the verifier rarely operates on a single still image; ownership is checked from short rendered multi-view clips. Motion, viewpoint changes, and rendering artifacts introduce frame-to-frame variations, so watermark evidence may be weak or inconsistent if decoding depends on individual

*Corresponding author.

frames. Second, to make watermarking scalable, we would like a reusable decoder that generalizes across scenes. A natural choice is to leverage CLIP [Radford *et al.*, 2021] to pre-train a message decoder under text supervision, but at verification time the evidence comes from rendered frames: CLIP visual embeddings inevitably entangle watermark traces with rich scene semantics (objects, textures, illumination), creating a distribution gap to the message-centric text embeddings used during decoder pre-training. Bridging this gap without sacrificing invisibility or robustness is non-trivial.

In this paper, we formulate watermarking for 4DGS for the first time. Given a reconstructed 4DGS model and a binary copyright message, the goal is to embed the message such that (i) normal renderings remain visually indistinguishable from the original, and (ii) the message can be reliably recovered from short rendered multi-view clips, without access to the original scene or model parameters (Fig. 1). To address the above challenges, we propose Guard4D, a decoupled watermarking framework. We first pre-train a general-purpose message decoder by exploiting CLIP's shared text–image embedding space: a deterministic bit-to-token mapping generates watermark token sequences, whose CLIP text embeddings supervise a lightweight bit decoder. This message decoder is trained once offline and then kept fixed across scenes.

For a target 4DGS asset, Guard4D embeds the message by optimizing compact additive offsets on the SH appearance coefficients while keeping geometry and opacity fixed. Because 4DGS rendering is fully differentiable, these offsets can be optimized end-to-end from rendered clips, making embedding efficient and reversible. To improve extraction from short clips and to mitigate the semantic entanglement in CLIP visual embeddings, Guard4D introduces a temporal modeling and semantic decoupling (TMSD) module placed before the frozen CLIP visual encoder. The module aggregates evidence across neighboring frames via 3D convolutions and transforms the rendered clip into a representation that emphasizes watermark before CLIP feature extraction, while keeping the downstream decoder unchanged. Moreover, semantic decoupling is enforced through supervision: we render paired clean and noisy control video segments from the original (non-watermarked) model under identical camera trajectories, and penalize message-consistent predictions on these semantic-only controls. This counterfactual regularization discourages semantic shortcuts and encourages the extractor to respond specifically to watermark-induced traces. We summarize our contributions as follows:

- We formulate 4DGS watermarking for the first time, targeting practical ownership verification by extracting a binary message directly from rendered multi-view videos of dynamic Gaussian scene representations.
- We propose Guard4D, a decoupled watermarking framework that embeds messages by optimizing SH offsets and enables clip-level extraction with a TMSD module optimized using clean and noisy control supervision to bridge the text–visual gap and suppress semantic interference.
- Experiments on dynamic 4DGS scenes demonstrate that Guard4D effectively suppresses semantic interference, enabling more accurate watermark recovery while preserving rendering fidelity.

2 Related Work

3D Gaussian Splatting and 3DGS Watermarking. Neural radiance fields (NeRF) [Mildenhall *et al.*, 2021] enable high-fidelity novel-view synthesis via implicit volumetric rendering, but their sampling-heavy formulation often incurs substantial computational overhead, motivating acceleration strategies [Wang *et al.*, 2023b] and generative extensions [Poole *et al.*, 2022; Wang *et al.*, 2023d]. To improve efficiency, 3DGS represents scenes with explicit anisotropic Gaussian primitives equipped with spherical-harmonic (SH) appearance features, enabling real-time rendering with high visual fidelity [Kerbl *et al.*, 2023]. Since its introduction, 3DGS has been extended into a versatile framework spanning reconstruction and geometry processing (e.g., mesh extraction and anti-aliasing) [Guédon and Lepetit, 2024; Yu *et al.*, 2024], SLAM and online mapping [Yan *et al.*, 2024; Matsuki *et al.*, 2024], as well as generative and animatable modeling [Liang *et al.*, 2024; Zou *et al.*, 2024; Shao *et al.*, 2024]. With the increasing adoption of neural rendering models in practical pipelines, protecting model ownership has become increasingly important. Classical digital watermarking methods were developed in the frequency domain [Navas *et al.*, 2008; Tao and Eskicioglu, 2004] and later advanced to deep learning based frameworks [Fernandez *et al.*, 2023; Wen *et al.*, 2023]. For 3D assets, prior works embed messages into explicit shapes or geometric structures [Zhang *et al.*, 2023; Zhu *et al.*, 2021]. Neural-scene watermarking approaches further embed information into NeRF models for copyright verification [Jang *et al.*, 2024; Song *et al.*, 2024b], but they are tied to implicit volumetric representations and cannot be directly applied to explicit 3DGS. Recent works therefore explore watermarking within Gaussian Splatting by injecting messages into the Gaussian parameter space and optimizing for invisibility and robustness under rendering [Huang *et al.*, 2024; Jang *et al.*, 2025; Chen *et al.*, 2025]. Geometry Cloak [Song *et al.*, 2024a] provides a complementary protection perspective by inducing identifiable reconstruction patterns via adversarial perturbations. Overall, existing 3DGS watermarking techniques operate exclusively on static scenes and typically assume time-invariant geometry and appearance.

Dynamic Gaussian Representations. Modeling dynamic scenes requires jointly capturing time-varying geometry, appearance, and motion. Early approaches extend static implicit representations to dynamic settings via deformation or canonical-space formulations [Gao *et al.*, 2022; Wang *et al.*, 2023c], yet they often remain computationally expensive and struggle with real-time performance. Explicit Gaussian-based dynamic representations provide a scalable alternative with compact primitives and fast rasterization. Dynamic 3D Gaussians [Luiten *et al.*, 2024] generalize 3DGS to time-varying scenes by allowing Gaussians to translate and rotate over time under local-rigidity regularization. Deformable 3D Gaussians [Yang *et al.*, 2024] further incorporate canonical-space deformation and differentiable rasterization for real-

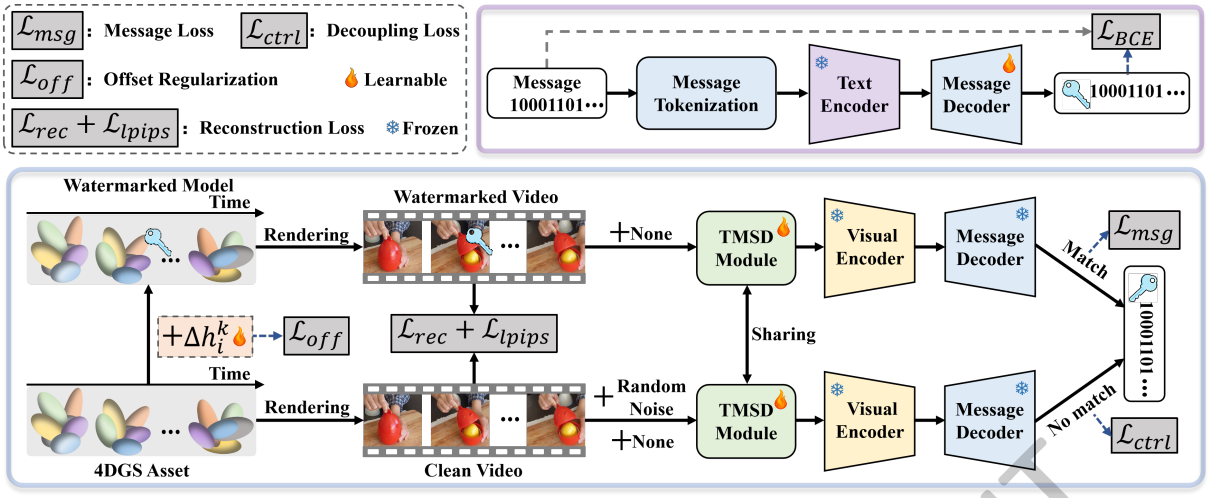

Figure 2: Overview of Guard4D. We pre-train a message decoder from CLIP text embeddings, then embed a binary message into a 4DGS asset by optimizing SH offsets and rendering multi-view watermarked videos with fidelity losses. For extraction, the shared TMSD module performs semantic decoupling on watermarked clips and clean/noisy control clips before the frozen CLIP visual encoder; the decoder is trained to match the target on watermarked clips ($\mathcal{L}_{\text{msg}}$) while discouraging message-aligned outputs on clean and noisy controls ($\mathcal{L}_{ctrl}$).

time reconstruction, while Spacetime Gaussian Feature Splatting [Li *et al.*, 2024] augments primitives with temporal features to better handle transient content. Parallel efforts explore explicit 4D representations and hybrid spatiotemporal structures for efficient video synthesis [Lin *et al.*, 2023; Wang *et al.*, 2023a]. Most relevant to our work, 4D Gaussian Splatting (4DGS) integrates temporally deformable Gaussians with neural voxels, enabling high-fidelity real-time rendering of complex motions [Wu *et al.*, 2024]. Despite rapid progress in dynamic reconstruction and rendering, watermarking for dynamic Gaussian representations remains largely unexplored: existing 3DGS watermarking methods provide no mechanisms to maintain watermark consistency under temporal deformation, motivating our study of robust and temporally stable watermarking for 4DGS.

3 Method

Our goal is to embed a binary copyright message into a reconstructed 4D Gaussian Splatting (4DGS) model such that the watermark is invisible in normal rendering and extractable from rendered multi-view clips without access to the original clean model. The proposed pipeline follows a decoupled design that separates a reusable message decoder from asset-specific embedding and extraction adaptation. After this offline pre-training stage, watermark embedding for a target 4DGS asset becomes an efficient optimization problem on a small set of learnable spherical-harmonic offsets while keeping the reconstructed geometry and opacity fixed. In addition, we introduce a lightweight temporal modeling and semantic decoupling (TMSD) module that improves extraction stability on short video clips by aggregating spatiotemporal cues and suppressing semantic interference.

3.1 4D Gaussian Splatting Preliminaries

4D Gaussian Splatting [Wu *et al.*, 2024] provides a dynamic scene by an explicit set of Gaussian primitives and renders them by differentiable rasterization. Each scene is represented as a set of anisotropic 3D Gaussian primitives $\mathcal{G}_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{h}_i\}$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the center, $\boldsymbol{\Sigma}_i$ is the covariance, α_i denotes the opacity, and $\mathbf{h}_i$ denotes the spherical harmonic (SH) appearance coefficients. The SH coefficients are stored in two groups, including the DC term and the remaining higher-order terms:

$$\mathbf{h}_i = \{\mathbf{h}_i^{\text{dc}}, \mathbf{h}_i^{\text{rest}}\}. \tag{1}$$

For a camera at time t with extrinsic transform W_t and projection P_t, the Gaussian mean and covariance are mapped to the image plane by $\hat{\boldsymbol{\mu}}_{i,t} = P_t W_t \boldsymbol{\mu}_{i,t}$, and $\hat{\boldsymbol{\Sigma}}_{i,t} = J_t W_t \boldsymbol{\Sigma}_{i,t} W_t^\top J_t^\top$, where J_t is the Jacobian of the local affine projection at $\hat{\boldsymbol{\mu}}_{i,t}$. Given a camera at time t, each Gaussian projects to an image-space ellipse, and its opacity contribution at pixel (x, y) is

$$\sigma_{i,t}(x, y) = \alpha_i \, \mathcal{N}\big((x, y); \hat{\boldsymbol{\mu}}_{i,t}, \hat{\boldsymbol{\Sigma}}_{i,t}\big), \tag{2}$$

where $\mathcal{N}(\cdot; \mu, \Sigma)$ denotes the 2D Gaussian density. Let $\mathbf{c}_{i,t}(x, y)$ be the view-dependent color obtained by evaluating SH coefficients $\mathbf{h}_i$ under the current viewing direction. The final pixel color is accumulated by depth-ordered alpha compositing:

$$\hat{\mathbf{C}}_t(x, y) = \sum_{i=1}^{N} \mathbf{c}_{i,t}(x, y)\, \sigma_{i,t}(x, y) \prod_{j<i} \big(1 - \sigma_{j,t}(x, y)\big). \tag{3}$$

This rendering process is fully differentiable, enabling us to optimize a small SH perturbation to embed a binary message while preserving appearance.

3.2 CLIP-guided Message Decoder Pre-training

We pre-train a lightweight message decoder D_ψ once using CLIP text features [Radford *et al.*, 2021], and then keep it fixed for all scenes. For each target 4DGS asset, only the

scene-dependent SH offsets and the TMSD are optimized. We leverage CLIP as a bridge between discrete watermark tokens and continuous visual representations. CLIP provides a text encoder that maps a token sequence to a 512-dimensional feature and a visual encoder that maps rendered frames into the same latent space. Instead of training a scene-specific image-to-message decoder, we first learn a general message decoder that predicts bits from CLIP features under text supervision. At verification time, the owner applies this fixed decoder together with the private TMSD front-end to rendered clips, without requiring access to the original clean 4DGS model.

Message tokenization. Let $\mathbf{m} \in \{0,1\}^L$ denote the target message, where L is the message length. We construct a CLIP token sequence $\mathbf{t} \in \mathbb{Z}^{77}$ by mapping each bit position to one of two reserved token IDs. Concretely, we pre-sample a bit-to-token table $\mathbf{\Phi} \in \mathbb{Z}^{L\times 2}$ and set the $(i+1)$-th token to Φ_{i,m_i}, while using special begin and end tokens at the boundaries. The resulting sequence is padded to CLIP's fixed length. This mapping is fixed once generated and saved, which makes the tokenization deterministic across runs.

Text-feature supervision and decoder learning. Given tokenized message $\mathbf{t}$, the CLIP text encoder produces a normalized feature $\mathbf{f}_T \in \mathbb{R}^{512}$. We train a lightweight message decoder D_ψ implemented as a multi-layer perceptron (MLP) to reconstruct the original message:

$$\hat{\mathbf{m}} = D_\psi(\mathbf{f}_T), \qquad \mathbf{f}_T = E_T(\mathbf{t}). \tag{4}$$

We supervise the logits $\hat{\mathbf{m}}$ with the standard bit-wise binary cross entropy:

$$\mathcal{L}_{\mathrm{dec}}(\psi) = \frac{1}{L}\sum_{i=1}^{L} \mathrm{BCEWithLogits}\big(\hat{m}_i, m_i\big). \tag{5}$$

During pre-training, all CLIP parameters are frozen and only the MLP decoder is optimized. This stage does not involve any 3D scene or rendering process, which makes it fast and scalable. After convergence, we keep the message decoder fixed. During watermark embedding and verification, it is used together with the asset-specific TMSD module as the private extractor.

3.3 4D Watermark Embedding and Differentiable Video Rendering

We embed the message into a reconstructed 4D Gaussian Splatting (4DGS) model by learning compact additive offsets on its stored SH coefficients, and then render multi-view video clips from the resulting watermarked model for extraction and verification. This formulation matches the practical setting where the evidence is a short rendered video rather than a single still image.

SH-offset parameterization for watermark embedding. Let $\mathcal{G}$ denote the original 4DGS model parameters and $\tilde{\mathcal{G}}$ denote the watermarked model. We freeze all Gaussian attributes except the spherical harmonic (SH) coefficients, and introduce trainable offsets on the SH buffers. For each Gaussian i and each SH block $k \in \{\mathrm{dc}, \mathrm{rest}\}$, we define

$$\tilde{\mathbf{h}}_i^k = \mathbf{h}_i^k + \Delta\mathbf{h}_i^k. \tag{6}$$

All offsets are stacked into tensors $\Delta\mathbf{H}^{\mathrm{dc}}$ and $\Delta\mathbf{H}^{\mathrm{rest}}$ whose shapes match the stored fields in the 4DGS implementation. We create a copy of the reconstructed Gaussian model and only modify the SH buffers of the copy, which keeps the original reconstruction intact and makes the watermarking process reversible by simply discarding the offsets. This embedding strategy binds the message to the scene representation rather than to a particular view. The offsets in Eq. (6) only affect the view-dependent color evaluation in the 4DGS renderer (Eq. (3)), and thus the watermark signal is distributed over many splats through the differentiable rasterizer. Because 4DGS rendering is consistent across novel viewpoints and time, the same SH offsets induce correlated watermark evidence across multi-view video clips, which is essential for stable extraction from short clips.

Offset control and regularization. The two SH blocks contribute differently to appearance. The DC component largely determines low-frequency color and global tone, while the higher-order block mainly influences view-dependent effects and high-frequency details. To control artifacts, we optimize $\Delta\mathbf{H}^{\mathrm{dc}}$ and $\Delta\mathbf{H}^{\mathrm{rest}}$ with different learning rates and apply stronger regularization to keep the perturbation small.

Differentiable rendering of watermarked videos. After injecting the watermark into $\tilde{\mathcal{G}}$, we render multi-view video clips using the standard 4DGS differentiable rasterizer. Let $\mathcal{C} = \{v\}_{v=1}^{V}$ denote a set of sampled camera viewpoints. Each view v is associated with a temporal sequence of frames indexed by $t \in \{1,\dots,F\}$. Rendering a clip from a 4DGS model produces a tensor $\mathbf{I} \in [0,1]^{V\times F\times 3\times H\times W}$, where H and W are the spatial resolution. We use $\mathbf{I}$ to denote reference renderings from the original model $\mathcal{G}$ and $\tilde{\mathbf{I}}$ to denote renderings from the watermarked model $\tilde{\mathcal{G}}$.

Stochastic multi-view temporal sampling. Since rendering every view and every frame is expensive, we adopt stochastic sampling during optimization. At each epoch, we sample a subset of viewpoints and then sample a small number of frames per view to form a mini-batch clip. This mini-batch still contains multi-view temporal redundancy while keeping the optimization cost low.

3.4 Message Decoder with Temporal Modeling and Semantic Decoupling

We consider the practical verification setting where the evidence is a short rendered clip. Let $\tilde{\mathbf{I}} \in [0,1]^{B\times D\times 3\times H\times W}$ denote a batch of clips rendered from the watermarked 4DGS model, and $\mathbf{m} \in \{0,1\}^L$ be the fixed target message. Compared with still images, clips may exhibit frame-to-frame variations caused by motion, viewpoint changes, and rendering noise, which can make decoding from any single frame less reliable. Since the embedded message is constant over the clip, aggregating information across multiple frames is expected to improve robustness in practice. A second challenge stems from the mismatch between how the message decoder is supervised and what is observable at verification time. Our decoder D_ψ is pre-trained under text supervision: it takes the CLIP text embedding computed from watermark

Method	D-NeRF (Synthetic)				HyperNeRF (Real-world)			
	Bit Acc ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Bit Acc ↑	PSNR ↑	SSIM ↑	LPIPS ↓
GuardSplat [Chen *et al.*, 2025]	88.64	41.17	0.9958	0.0036	**97.75**	42.97	0.9973	0.0032
Baseline	89.38	42.75	0.9969	0.0011	76.55	46.37	0.9984	0.0013
Guard4D (Ours)	**99.81**	**49.87**	**0.9992**	**0.0002**	96.83	**54.19**	**0.9995**	**0.0003**

Table 1: Quantitative comparison on D-NeRF and HyperNeRF with a fixed 16-bit watermark. Results are reported for single-frame GuardSplat, multi-frame GuardSplat (baseline), and the proposed Guard4D with semantic decoupling.

tokens (Sec. 3.2), yielding a message-focused representation. At verification time, however, we only observe rendered frames. Encoding these frames with the frozen CLIP visual encoder produces embeddings that inevitably mix watermark traces with scene semantics (e.g., objects, textures, illumination), creating a distribution gap relative to the message-centric text embeddings used in pre-training. We address these issues with a unified TMSD module that aggregates evidence over time, while semantic decoupling is primarily enforced through supervision using clean and noisy control clips.

Spatiotemporal watermark feature aggregation. We introduce a TMSD module R_ϕ placed in front of CLIP to suppress semantic interference. Given a clip, we apply a residual 3D ConvNet:

$$\tilde{\mathbf{I}}^\star = R_\phi(\tilde{\mathbf{I}}) = \tilde{\mathbf{I}} + s \cdot \Delta_\phi(\tilde{\mathbf{I}}), \tag{7}$$

where Δ_ϕ is a shallow 3D convolutional network and s is a residual scaling factor. The 3D kernels span the temporal axis, allowing each output frame to incorporate information from neighboring frames and consolidating multi-frame evidence. The resulting semantics-suppressed clip is then encoded by the frozen CLIP visual encoder E_V and decoded by the pre-trained message decoder:

$$\mathbf{o} = D_\psi\Big(E_V(\tilde{\mathbf{I}}^\star)\Big), \qquad \hat{\mathbf{m}} = \sigma(\mathbf{o}), \tag{8}$$

where $\mathbf{o}$ denotes bit logits and $\sigma(\cdot)$ is the sigmoid.

Supervision for semantic decoupling with clean/noisy controls. Semantic decoupling in our design is imposed primarily through supervision. In addition to the watermarked clip $\tilde{\mathbf{I}}$ rendered from $\tilde{\mathcal{G}}$, we render a paired clean clip $\mathbf{I}$ from the original (non-watermarked) model $\mathcal{G}$ under the same camera trajectory. By construction, $\mathbf{I}$ contains the same scene semantics as $\tilde{\mathbf{I}}$ but does not contain the target watermark signal. We further form a perturbed variant $\mathbf{I}^n = \mathbf{I} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$. These two clips provide explicit counterfactual inputs: they allow us to regularize the extractor so that message-aligned evidence arises from watermark traces rather than from semantic content, while the noisy variant prevents the regularization from being satisfied by fragile, noise-sensitive cues.

Let $\mathbf{m}$ denote the target message. We obtain the main logits $\mathbf{o}$ from the semantically decoupled $\tilde{\mathbf{I}}^\star = R_\phi(\tilde{\mathbf{I}})$, and compute two auxiliary logits $\mathbf{o}^{orig}$ and $\mathbf{o}^{noisy}$ by running the same $R_\phi \to E_V \to D_\psi$ pipeline on $\mathbf{I}$ and $\mathbf{I}^n$, respectively. We train the main branch with the standard bit-wise binary cross entropy:

$$\mathcal{L}_{\text{msg}} = \text{BCEWithLogits}(\mathbf{o}, \mathbf{m}). \tag{9}$$

To enforce semantic decoupling, we introduce an inverse-BCE regularizer on the two control branches:

$$\begin{aligned}\mathcal{L}_{\text{ctrl}} &= \mathcal{L}_{ctrl_o} + \mathcal{L}_{ctrl_n} \\ &= \frac{1}{\text{BCEWithLogits}(\mathbf{o}^{orig}, \mathbf{m}) + \varepsilon} \\ &\quad + \frac{1}{\text{BCEWithLogits}(\mathbf{o}^{noisy}, \mathbf{m}) + \varepsilon}.\end{aligned} \tag{10}$$

3.5 Training Objective

Let $\tilde{\mathbf{I}} \in [0,1]^{B\times D\times 3\times H\times W}$ denote a batch of clips rendered from the watermarked model $\tilde{\mathcal{G}}$, and let $\mathbf{I}$ denote the paired clean clips rendered from the original model $\mathcal{G}$ under the same sampled cameras and frames. The target watermark is a binary vector $\mathbf{m} \in \{0,1\}^L$ to be decoded from rendered video clips.

Private decoding pipeline. Following Sec. 3.4, verification uses a private decoder that applies a learnable TMSD module before CLIP feature extraction:

$$D(\mathbf{J}) \;=\; D_\psi(E_V(R_\phi(\mathbf{J}))) \in \mathbb{R}^L, \tag{11}$$

where R_ϕ is the TMSD module for watermark-evidence enhancement, E_V is the (frozen) CLIP visual encoder, and D_ψ is the pre-trained message decoder. In our default training, we optimize the watermark offsets together with ϕ (and any additional enhancer parameters, collectively denoted as θ when needed), while keeping E_V fixed; D_ψ is also kept fixed unless explicitly stated. Overall, we balance three goals: (i) the rendered appearance of the watermarked model should remain close to the original; (ii) the private decoder should reliably recover $\mathbf{m}$ from watermarked renders; (iii) the decoder should not spuriously recover $\mathbf{m}$ from clean renders that share the same scene semantics.

Reconstruction loss for visual fidelity. To preserve the usability of the 4DGS asset, watermarked renders should be visually close to the original renders. We measure fidelity by combining a pixel-wise term with perceptual similarity:

$$\mathcal{L}_{\text{rgb}} = (1-\lambda_{\text{ssim}})\,\|\tilde{\mathbf{I}} - \mathbf{I}\|_1 + \lambda_{\text{ssim}}\big(1 - \text{SSIM}(\tilde{\mathbf{I}}, \mathbf{I})\big), \tag{12}$$

where $\text{SSIM}(\cdot,\cdot)$ is the differentiable structural similarity index. We further incorporate a learned perceptual metric:

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{lpips}}\,\text{LPIPS}(\tilde{\mathbf{I}}, \mathbf{I}), \tag{13}$$

where $\text{LPIPS}(\cdot,\cdot)$ is computed by a frozen network [Zhang *et al.*, 2018]. This term discourages visually noticeable artifacts and encourages the watermark signal to distribute in a perceptually less salient manner.

Message Length	D-NeRF (Synthetic)				HyperNeRF (Real-world)			
	Bit Acc ↑	PSNR ↑	SSIM ↑	LPIPS ↓	Bit Acc ↑	PSNR ↑	SSIM ↑	LPIPS ↓
16 bits	99.81	49.87	0.9992	0.0002	96.83	54.19	0.9995	0.0003
32 bits	94.26	48.28	0.9987	0.0004	86.17	50.49	0.9989	0.0008
48 bits	89.95	46.98	0.9988	0.0005	86.77	50.72	0.9991	0.0008

Table 2: Ablation study on watermark message length on the D-NeRF and HyperNeRF datasets.

Offset regularization. Even with reconstruction losses, unconstrained watermark offsets may accumulate and degrade fidelity. We therefore penalize the magnitude of the offsets with an ℓ_2 regularizer:

$$\mathcal{L}_{\text{off}} = \|\Delta \mathbf{H}^{\text{dc}}\|_2^2 + \|\Delta \mathbf{H}^{\text{rest}}\|_2^2. \tag{14}$$

This term prevents excessive modifications of spherical harmonic coefficients and improves both invisibility and stability.

Overall objective. Combining the above terms, we optimize the following objective:

$$\mathcal{L} = \lambda_{\text{recon}}\,\mathcal{L}_{\text{recon}} + \lambda_{\text{msg}}\,(\mathcal{L}_{\text{msg}} + \mathcal{L}_{\text{ctrl}}) + \lambda_{\text{off}}\,\mathcal{L}_{\text{off}}. \tag{15}$$

4 Experiments

4.1 Dataset

We evaluate the performance of our proposed Guard4D framework on both synthetic and real-world dynamic scene datasets, following standard evaluation protocols used in dynamic radiance field research. Specifically, we adopt the D-NeRF [Pumarola *et al.*, 2021] and HyperNeRF [Park *et al.*, 2021] datasets for quantitative and qualitative analysis. The D-NeRF dataset contains eight synthetic dynamic scenes with complex non-rigid motions and randomly generated camera trajectories, serving as a controlled benchmark for assessing temporal consistency and rendering fidelity under known geometry and motion. In contrast, the HyperNeRF dataset consists of four real captured dynamic scenes that exhibit topological and motion variations under unconstrained illumination and camera settings, providing a challenging evaluation of robustness in real-world conditions. For each scene from both datasets, we render an 8-frame video sequence with 100 camera viewpoints per frame to comprehensively evaluate visual quality and watermark bit accuracy. We report the averaged results over all testing views of each scene in our experiments.

4.2 Evaluation Metrics

We assess the performance of Guard4D from both watermarking and rendering perspectives using four quantitative metrics, following standard evaluation protocols in digital watermarking and neural radiance field rendering. Specifically, we evaluate the Bit Accuracy (Bit Acc), which measures the proportion of correctly extracted watermark bits relative to the embedded message and reflects the robustness and reliability of watermark recovery under different temporal frames and camera viewpoints. For rendering quality, we employ the Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index (SSIM) [Wang *et al.*, 2004], and the Learned Perceptual Image Patch Similarity (LPIPS) [Zhang *et al.*, 2018] to evaluate the visual similarity between rendered video frames from the watermarked and original 4D Gaussian Splatting models. PSNR quantifies pixel-wise reconstruction fidelity, SSIM measures the structural consistency of luminance and contrast, and LPIPS provides a perceptual assessment of visual quality based on deep feature similarity. Higher PSNR and SSIM values, together with lower LPIPS scores, indicate better visual quality and invisibility of the embedded watermark.

4.3 Implementation Details

We train Guard4D for 100 epochs on a single NVIDIA A100 GPU with a batch size of 8. Watermark embedding is formulated as an optimization over additive offsets to the spherical harmonic coefficients of a fixed 4D Gaussian Splatting model, while all other Gaussian parameters are kept frozen. We jointly optimize the SH offsets and the TMSD module using the Adam optimizer, with a learning rate of 5×10^{-3} for the DC SH components and a $20\times$ smaller learning rate for the higher-order SH coefficients. The optimization objective consists of a reconstruction loss, a message extraction loss, a control loss for semantic decoupling, and an offset regularization term. The message and control losses share the weight $\lambda_{\text{msg}} = 0.03$, with $\lambda_{\text{recon}} = 1$ and $\lambda_{\text{off}} = 10$.

4.4 Comparison with Other Methods

Table 1 presents a quantitative comparison under a fixed 16-bit watermark message length on both the D-NeRF and HyperNeRF datasets. The results reveal a clear performance evolution from single-frame GuardSplat to the multi-frame training baseline, and finally to our full Guard4D framework with semantic decoupling. On the synthetic D-NeRF dataset, extending the training from single-frame inputs to multi-frame video clips improves rendering fidelity, increasing PSNR from 41.17 dB to 42.75 dB and reducing LPIPS from 0.0036 to 0.0011. However, the gain in watermark robustness remains limited, with bit accuracy improving by less than one percentage point. By introducing semantic decoupling, Guard4D boosts bit accuracy from 89.38% to 99.81%, while further increasing PSNR to 49.87 dB and reducing LPIPS to 0.0002. On the real-world HyperNeRF dataset, the multi-frame training baseline improves rendering quality but suffers a severe degradation in watermark extraction accuracy, with bit accuracy dropping from 97.75% to 76.55%. This result suggests that temporal aggregation alone may amplify spatiotemporal scene semantics rather than water-

$\mathcal{L}_{\text{msg}}$	$\mathcal{L}_{\text{recon}}$	$\mathcal{L}_{off}$	$\mathcal{L}_{\text{ctrl}}$	Bit Acc ↑	PSNR ↑	SSIM ↑	LPIPS ↓
				52.52	inf	1.0000	0.0000
✓		✓	✓	99.92	39.96	0.9950	0.0098
✓	✓		✓	99.97	41.67	0.9365	0.0056
✓	✓	✓		100.00	76.15	0.9999	0.0000
✓	✓	✓	✓	99.81	49.87	0.9992	0.0002

Table 3: Ablation on loss function components on D-NeRF under a fixed 16-bit watermark message.

$\mathcal{L}_{\text{msg}}$	$\mathcal{L}_{\text{ctrl}_o}$	$\mathcal{L}_{\text{ctrl}_n}$	WM videos ↑	Orig. videos ↓	Noisy videos ↓
✓			100.00	96.45	97.86
✓		✓	99.97	52.34	51.34
✓	✓		99.89	49.26	49.67
✓	✓	✓	99.81	47.39	46.36

Table 4: Ablation on control loss components on D-NeRF under a fixed 16-bit watermark message. WM, Orig., and Noisy denote watermarked, original, and noise-perturbed videos, respectively.

mark evidence in complex real captured scenes. In contrast, Guard4D restores robust watermark extraction, improving bit accuracy by more than 20 percentage points over the multi-frame training baseline, while increasing PSNR from 46.37 dB to 54.19 dB and reducing LPIPS from 0.0013 to 0.0003. Overall, these results demonstrate that simply increasing temporal context is insufficient for robust watermarking in dynamic scenes. Semantic decoupling plays a critical role in stabilizing message extraction while preserving high visual fidelity. Guard4D achieves a better trade-off between watermark robustness and rendering quality, making it well suited for protecting dynamic 4D Gaussian Splatting assets.

4.5 Ablation Study

Ablation on Watermark Message Length. Table 2 evaluates the effect of watermark message length by varying the payload from 16 to 48 bits. As the message length increases, watermark extraction accuracy decreases gradually on both datasets, dropping from 99.81% to 89.95% on D-NeRF and from 96.83% to 86.77% on HyperNeRF, reflecting the expected trade-off between watermark capacity and robustness. Despite the increased payload, Guard4D consistently maintains high rendering quality across all settings, with PSNR remaining above 46 dB on D-NeRF and 50 dB on HyperNeRF, SSIM consistently above 0.998, and LPIPS staying at very low levels. These results indicate that Guard4D supports flexible watermark capacity with controlled degradation in robustness and negligible impact on visual fidelity.

Ablation on Loss Function Components. We analyze the effect of different loss terms in our training objective in Table 3. The results show that no single loss term alone is sufficient to achieve both high watermark robustness and good visual fidelity. Without the message loss, the model preserves the original rendering but fails to embed a recoverable watermark, yielding near-random bit accuracy.

Introducing the reconstruction loss $\mathcal{L}_{\text{recon}}$ and the offset regularization $\mathcal{L}_{off}$ significantly improves visual quality, as reflected by increased PSNR and reduced LPIPS. However, without the control loss $\mathcal{L}_{\text{ctrl}}$, the decoder can trivially recover the message by exploiting scene-dependent cues, leading to unrealistically high bit accuracy accompanied by overfitting and unstable generalization. By incorporating $\mathcal{L}_{\text{ctrl}}$, the model achieves a balanced solution with consistently high watermark extraction accuracy while maintaining strong rendering fidelity. The full loss combination yields the best overall trade-off, confirming that $\mathcal{L}_{\text{ctrl}}$ is essential for preventing spurious message recovery and enforcing that the decoded message originates from the embedded watermark rather than scene semantics.

Ablation on Control Loss Components. We further isolate the role of $\mathcal{L}_{\text{ctrl}}$ by ablating its internal components while keeping all other loss terms enabled, as shown in Table 4. Without any control term, the decoder exhibits severe semantic leakage, achieving high apparent accuracy not only on watermarked videos but also on original and noisy videos. Adding either the clean-video control term $\mathcal{L}_{ctrl_o}$ or the noisy-video control term $\mathcal{L}_{ctrl_n}$ alone partially suppresses false-positive predictions, but significant message recovery still occurs on non-watermarked inputs. Only when both $\mathcal{L}_{ctrl_o}$ and $\mathcal{L}_{ctrl_n}$ are jointly applied does the decoder achieve robust extraction on watermarked videos while effectively suppressing message recovery from original and noisy videos. This result confirms that the two control branches are complementary: $\mathcal{L}_{ctrl_o}$ discourages reliance on static scene semantics, while $\mathcal{L}_{ctrl_n}$ prevents the model from exploiting fragile, noise-sensitive cues. Together, they enforce effective semantic decoupling, making $\mathcal{L}_{\text{ctrl}}$ a critical component of the Guard4D framework.

We also evaluate robustness to common post-processing perturbations on rendered clips, including noise, rotation, scaling, blur, cropping, brightness adjustment, and JPEG compression. Guard4D shows only limited degradation in decoding performance under these perturbations, demonstrating robustness consistent with that of the GuardSplat [Chen *et al.*, 2025] baseline under image-level distortions.

5 Conclusion

We presented Guard4D, the first watermarking framework tailored to 4D Gaussian Splatting assets. Different from prior 3DGS watermarking methods that assume static scenes, Guard4D targets spatiotemporally deformable Gaussian fields and enables practical ownership verification from short rendered multi-view clips. Our method keeps the reconstructed 4DGS geometry fixed and embeds a binary message by optimizing compact residual offsets on SH appearance features, preserving rendering fidelity while enabling consistent watermark recovery across dynamic views. We further improve clip-level stability by using a lightweight TMSD to enhance watermark features and suppress semantic interference through paired clean and noisy control renderings. Experiments on synthetic and real-world dynamic scenes demonstrate that Guard4D effectively suppresses semantic interference, leading to accurate watermark recovery. Future work will explore extending Guard4D to broader dynamic representations and stronger security guarantees against adaptive removal and model editing for practical protection of dynamic neural assets.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and suggestions, which helped improve the paper. This work was supported in part by the National Natural Science Foundation of China (U25A20441), the National Science and Technology Major Project of China (2025ZD1700704), the National Key Research and Development Program of China (2024YFB3311703), and Zhongguancun Laboratory.

References

[Boboc *et al.*, 2022] Răzvan Gabriel Boboc, Elena Băutu, Florin Gîrbacia, Norina Popovici, and Dorin-Mircea Popovici. Augmented reality in cultural heritage: an overview of the last decade of applications. *Applied Sciences*, 12(19):9859, 2022.

[Boenisch, 2021] Franziska Boenisch. A systematic review on model watermarking for neural networks. *Frontiers in big Data*, 4:729663, 2021.

[Chen *et al.*, 2023] Lifeng Chen, Jia Liu, Yan Ke, Wenquan Sun, Weina Dong, and Xiaozhong Pan. Marknerf:watermarking for neural radiance field. *arXiv preprint arXiv:2309.11747*, 2023.

[Chen *et al.*, 2025] Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie. Guardsplat: Efficient and robust watermarking for 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16325–16335, 2025.

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

[Gao *et al.*, 2022] Xiangjun Gao, Jiaolong Yang, Jongyoo Kim, Sida Peng, Zicheng Liu, and Xin Tong. Mps-nerf: Generalizable 3d human rendering from multiview images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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

[Huang *et al.*, 2024] Xiufeng Huang, Ruiqi Li, Yiu-ming Cheung, Ka Chun Cheung, Simon See, and Renjie Wan. Gaussianmarker: Uncertainty-aware copyright protection of 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:33037–33060, 2024.

[Jang *et al.*, 2024] Youngdong Jang, Dong In Lee, MinHyuk Jang, Jong Wook Kim, Feng Yang, and Sangpil Kim. Waterf: Robust watermarks in radiance fields for protection of copyrights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12087–12097, 2024.

[Jang *et al.*, 2025] Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3d-gsw: 3d gaussian splatting for robust watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5938–5948, 2025.

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

[Li *et al.*, 2024] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024.

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

[Lin *et al.*, 2023] Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. High-fidelity and real-time novel view synthesis for dynamic scenes. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–9, 2023.

[Luiten *et al.*, 2024] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024.

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

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

[Narendra *et al.*, 2022] Modigari Narendra, ML Valarmathi, and L Jani Anbarasi. Watermarking techniques for three-dimensional (3d) mesh models: a survey. *Multimedia systems*, 28(2):623–641, 2022.

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

[Park *et al.*, 2021] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

[Poole *et al.*, 2022] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[Pumarola *et al.*, 2021] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

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

[Song *et al.*, 2024a] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. Geometry cloak: Preventing tgs-based 3d reconstruction from copyrighted images. *Advances in Neural Information Processing Systems*, 37:119361–119385, 2024.

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

[Tao and Eskicioglu, 2004] Peining Tao and Ahmet M Eskicioglu. A robust multiple watermarking scheme in the discrete wavelet transform domain. In *Internet Multimedia Management Systems V*, volume 5601, pages 133–144. SPIE, 2004.

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Wang *et al.*, 2023a] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023.

[Wang *et al.*, 2023b] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9065–9076, 2023.

[Wang *et al.*, 2023c] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023.

[Wang *et al.*, 2023d] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

[Wen *et al.*, 2023] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36:58047–58063, 2023.

[Wu *et al.*, 2024] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024.

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

[Yang *et al.*, 2024] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

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

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[Zhang *et al.*, 2023] Yushu Zhang, Jiahao Zhu, Mingfu Xue, Xinpeng Zhang, and Xiaochun Cao. Adaptive 3d mesh steganography based on feature-preserving distortion. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):5299–5312, 2023.

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

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