GUARDMARKGS: UNIFIED OWNERSHIP TRACING AND EDIT DETERRENCE FOR 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) is becoming a practical representation for novel view synthesis, but its growing adoption, together with rapid advances in instruction-driven 3DGS editing, also exposes a dual copyright risk: once released, a 3DGS-based asset can be used without permission and manipulated through 3D editing. Existing protection methods address only one side of this problem. Watermarking can trace ownership after unauthorized use, but it cannot prevent malicious editing. Adversarial edit-deterrence methods can disrupt editing, but they do not provide evidence of ownership. To the best of our knowledge, we present the first unified protection framework for 3DGS that jointly optimizes ownership tracing and unauthorized editing deterrence. Our framework combines a scene-wide watermarking objective over all Gaussians with an adversarial edit-deterrence objective. The adversarial branch combines latent-anchor separation, denoising-trajectory diversion, and cross-attention diversion to divert the editing trajectory, while an update-saliency-motivated Gaussian selection strategy assigns stronger adversarial updates to mask-selected Gaussians, improving the balance among watermark recovery, edit deterrence, and rendering fidelity. Experiments on scenes from Mip-NeRF 360 and Instruct-NeRF2NeRF demonstrate that the framework achieves a favorable balance among bit accuracy, edit deterrence, and rendering quality. These results suggest that practical copyright protection of 3DGS-based assets can be more effectively addressed by integrating ownership tracing and unauthorized editing deterrence into a single optimization framework.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as a powerful representation for novel view synthesis (Mildenhall et al., 2020; Müller et al., 2022; Lu et al., 2024), combining photo-realistic rendering quality with real-time rendering speed. As 3DGS assets are increasingly used in AR/VR, digital content production, and other commercial pipelines, copyright protection becomes a central concern. In practice, a released 3DGS scene can be redistributed without authorization and manipulated by 3D editing pipelines (Haque et al., 2023; Chen et al., 2024a;b; Wu et al., 2024; Lee et al., 2025). Recent advances in text-driven editing (Wang et al., 2022; Zhuang et al., 2023; Sella et al., 2023) further make it possible to produce visually convincing derivatives from released assets using only text prompts. As illustrated in Figure 1, this creates a practical threat scenario in which an owner may lose control over both proving ownership and preserving the integrity of the content.

These observations indicate that copyright protection for 3DGS should not be formulated only as watermark insertion after release. A watermark can support ownership tracing after unauthorized use, but it cannot prevent the released asset from being transformed into a semantically altered derivative. A practical protection mechanism should therefore preserve evidence of ownership while also deterring unauthorized instruction-driven editing.

Existing protection strategies are split into two disconnected families. Passive protection methods, such as ownership tracing via digital watermarking (Zhu et al., 2018; Fernandez et al., 2023; Wen et al., 2023; Huang et al., 2024b; Li et al., 2023; Luo et al., 2023; Jang et al., 2024), identify an asset's source after unauthorized use or redistribution, but their role is fundamentally post-hoc. By contrast, active protection methods based on adversarial optimization (Salman et al., 2023; Liang et al., 2023; Jeon et al., 2025; Liang & Wu, 2023; Shan et al., 2023; Meng et al., 2025) aim to make

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

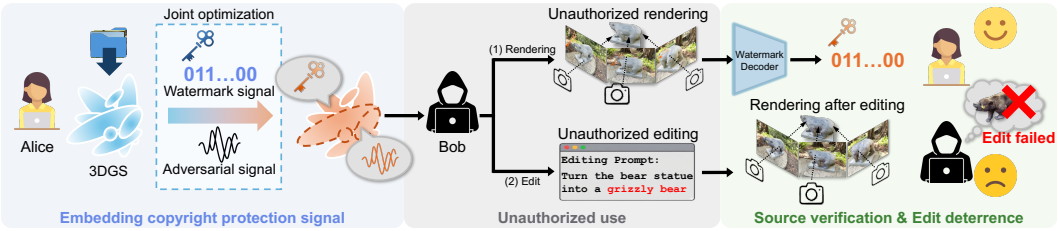


Figure 1: **Application scenario of GuardMarkGS.** We aim to identify ownership and prevent unauthorized editing by embedding watermark signals and adversarial noise into a 3DGS asset. Left: Alice optimizes her asset with watermark and adversarial signals. Middle: Bob performs unauthorized rendering and editing using Alice's asset. Right: Alice verifies ownership by recovering the watermark with her decoder, while preventing Bob from obtaining the intended editing results.

unauthorized editing difficult, but they do not embed a recoverable ownership signal for verification. Thus, watermark-only methods focus on traceability, whereas adversarial edit-deterrence methods focus on deterrence, and neither alone fully addresses the practical copyright threat in Figure 1.

However, unifying these two protection goals for 3DGS remains largely unexplored and is particularly challenging in the 3D setting. Reliable watermark recovery usually benefits from scene-wide optimization, because the ownership signal should remain detectable from many rendered viewpoints (Jang et al., 2025; Huang et al., 2024c; Chen et al., 2025). In contrast, imposing adversarial signals uniformly on the same scene-wide Gaussian support can unnecessarily perturb regions that are important for watermark recovery and rendering quality. This issue is not resolved by a straightforward composition of existing objectives: as shown in our experiments, directly adding DEGauss-style adversarial terms (Meng et al., 2025) to a 3DGSW-style watermarking objective (Jang et al., 2025) substantially weakens watermark recovery and rendering fidelity while providing limited edit deterrence. Thus, successful joint optimization must account for the different spatial and functional characteristics of watermarking and adversarial edit deterrence.

To address this joint-optimization conflict, we propose **GuardMarkGS**, a unified protection framework that jointly optimizes ownership tracing and unauthorized editing deterrence within a single protected 3DGS. GuardMarkGS assigns distinct roles to the two protection signals within a single coordinated optimization. The watermarking branch preserves watermark signal through a scene-wide objective over rendered views, while the adversarial branch assigns stronger edit-deterrence updates to mask-selected Gaussians identified by update-saliency-motivated selection. We further introduce parameter-role-aware adversarial modulation, which scales adversarial gradients according to the rendering sensitivity of each Gaussian parameter. This selective and role-aware allocation reduces unnecessary interference with watermark recovery and rendering fidelity, which is the key difficulty in naive joint optimization. To further disrupt unauthorized editing, the adversarial branch combines latent-anchor separation with denoising-trajectory and cross-attention diversion terms (Rombach et al., 2022), targeting the latent representation, denoising direction, and prompt-image alignment used by editors. This design provides a unified mechanism for balancing watermark recovery, edit deterrence, and rendering fidelity. Under sUCPS, a unified score that jointly evaluates traceability, edit deterrence, and fidelity, GuardMarkGS improves over the best existing baseline by 10.7%. Our main contributions are as follows:

- We propose a unified dual-branch framework that jointly optimizes scene-wide ownership tracing and unauthorized edit deterrence for a single protected 3DGS representation.
- We introduce update-saliency-motivated Gaussian selection and parameter-role-aware adversarial modulation, which allocate stronger adversarial updates to defense-effective Gaussians while reducing degradation to watermark recovery and rendering fidelity.
- We introduce an adversarial edit-deterrence objective that integrates latent-anchor separation with denoising-trajectory and cross-attention diversion, and experimentally show that the method achieves a sound balance among traceability, edit deterrence, and fidelity.

2 RELATED WORK

3D Gaussian Splatting representation. 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become a widely used representation for novel view synthesis by modeling a scene with explicit anisotropic Gaussian primitives and rendering them efficiently through differentiable splatting. Its combination of photorealistic quality and real-time rendering makes 3DGS increasingly practical for interactive 3D applications, digital content production, and downstream asset reuse (Yu et al., 2024a; Guédon & Lepetit, 2024; Huang et al., 2024a; Yu et al., 2024b; Tang et al., 2024; Yi et al., 2024). At the same time, the explicit Gaussian parameters make 3DGS directly editable and easy to redistribute. Once a 3DGS asset is released, its parameters can be copied, modified, or fine-tuned by downstream users, which creates a need for copyright protection mechanisms that preserve rendering fidelity while enabling ownership tracing of the released representation or protecting it against unauthorized manipulation.

Copyright protection for 3D Gaussian Splatting. Recent 3DGS copyright protection methods mostly follow a passive paradigm. GaussianMarker (Huang et al., 2024c), GuardSplat (Chen et al., 2025), and 3DGSW (Jang et al., 2025) embed watermark signals into Gaussian parameters while attempting to preserve rendering quality. These methods support post-hoc ownership tracing: after asset leakage or redistribution, owners can render protected-scene images and decode a message to verify provenance. However, they do not explicitly prevent malicious users from applying instruction-driven 3D edits (Haque et al., 2023; Chen et al., 2024a; Wu et al., 2024; Lee et al., 2025) to the released asset. Thus, they solve the tracing problem but not the deterrence problem: ownership can be verified after misuse, but the released content may already have been semantically altered.

Defenses against instruction-driven editing. A separate line of research studies active defenses (Salman et al., 2023; Liang et al., 2023; Jeon et al., 2025; Liang & Wu, 2023; Shan et al., 2023; Van Le et al., 2023) against diffusion-based image editing (Rombach et al., 2022; Brooks et al., 2023; Ho et al., 2020; Ho & Salimans, 2021). These methods inject adversarial perturbations that disrupt latent diffusion guidance or denoising trajectories, thereby lowering edit success. Extending this idea to 3D assets is challenging because 3D editing pipelines repeatedly render the current scene from multiple viewpoints, edit the rendered images, and optimize the 3D representation to match the edited views. Perturbations that are effective for a single image may not remain consistent across viewpoints or may be weakened during iterative 3D optimization. DEGauss (Meng et al., 2025) extends adversarial protection to 3DGS and is, to our knowledge, the most relevant defense-only baseline in the 3D setting. However, DEGauss does not provide ownership tracing, and its whole-scene adversarial protection can noticeably degrade rendering quality.

3D Gaussian Splatting editing pipelines. Text-driven 3DGS editing has advanced rapidly through pipelines such as DGE (Chen et al., 2024a) and GaussianEditor (Chen et al., 2024b). These methods repeatedly render the current 3D representation, edit the views with an image editing model (Brooks et al., 2023), and fit the 3DGS back to the edited results. This iterative render-edit-update cycle makes 3DGS editing powerful, but it also exposes a practical pathway for unauthorized manipulation. If an unauthorized user can access the released 3DGS model, they can produce high-quality malicious variants with only a text prompt. Thus, the same properties that make 3DGS useful for legitimate editing also make it vulnerable to misuse.

The above studies address important but incomplete aspects of 3DGS protection. Watermarking methods provide ownership evidence, but they do not actively deter unauthorized edits. Adversarial defense methods can suppress instruction-driven editing, but they do not provide recoverable ownership signals. Our work targets this missing intersection. Rather than treating ownership tracing and edit deterrence as separate goals, we formulate them as a unified optimization problem for a single protected 3DGS model. A scene-wide watermarking objective preserves ownership evidence across views, while an update-saliency-motivated adversarial objective allocates stronger updates to mask-selected Gaussian regions. This design provides a practical protection framework that supports ownership tracing, resists unauthorized editing, and simultaneously accounts for rendering quality.

3 PROBLEM SETUP

Let $\mathcal{G} = \{g_i\}_{i=1}^{N}$ denote a 3DGS scene composed of N Gaussians, where each Gaussian contains geometric and appearance parameters such as position, scale, rotation, opacity, and view-dependent

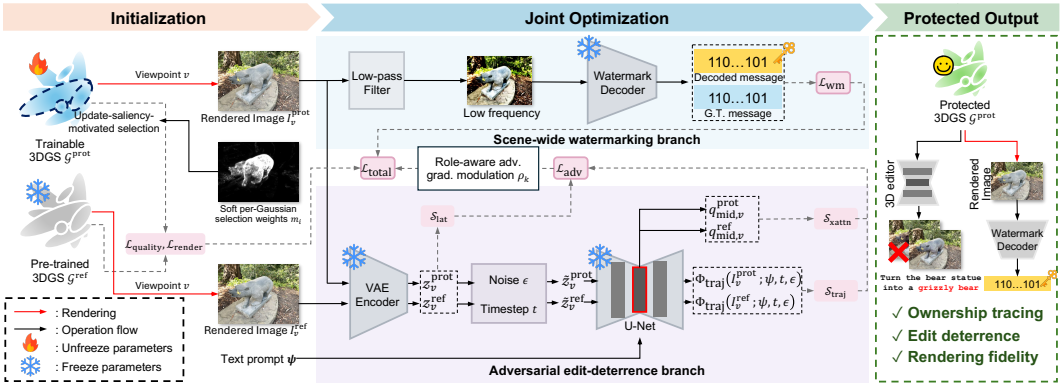


Figure 2: **Overview of GuardMarkGS.** Left: We initialize 3DGS from a pretrained scene, keep a reference, and compute soft per-Gaussian weights via update-saliency-motivated Gaussian selection. Middle: The watermarking branch embeds watermark signals, while the adversarial branch uses latent-anchor separation, denoising-trajectory diversion, and cross-attention diversion. These terms use rendering constraints and softly weighted adversarial gradients modulated by parameter role. Right: The optimized 3DGS preserves fidelity, supports ownership tracing, and deters editing.

features. Rendering the scene from a camera viewpoint v yields an image $I_v = \mathcal{R}_v(\mathcal{G})$, where $\mathcal{R}_v(\cdot)$ denotes the differentiable 3DGS renderer for viewpoint v.

Our goal is to optimize a protected scene $\mathcal{G}^{\text{prot}}$ that satisfies three requirements simultaneously. First, it must carry an ownership message $\mathbf{m} \in \{0,1\}^k$, where k denotes the message length, that can be decoded from rendered views with high bit accuracy. Second, under the threat scenario illustrated in Figure 1, in which the released asset is subjected to editing with an edit instruction ψ, the edited outputs should deviate from the ideal editing results. Third, it must preserve the visual quality of the original scene in novel-view rendering. Formally, the protected scene should satisfy

$$\text{Tracing:} \quad \mathcal{D}_{\text{msg}}\big(\mathcal{R}_v(\mathcal{G}^{\text{prot}})\big) \to \mathbf{m}, \tag{1}$$

$$\text{Deterrence:} \quad \mathcal{E}\big(\mathcal{G}^{\text{prot}}, \psi\big) \not\approx \mathcal{E}(\mathcal{G}, \psi), \tag{2}$$

$$\text{Fidelity:} \quad \mathcal{R}_v(\mathcal{G}^{\text{prot}}) \approx \mathcal{R}_v(\mathcal{G}), \tag{3}$$

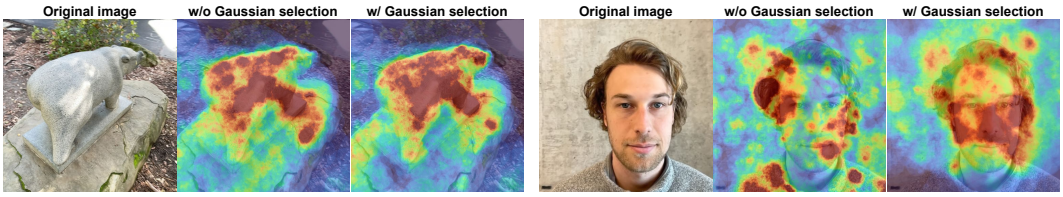
where $\mathcal{D}_{\text{msg}}$ is the watermark decoder, $\mathcal{E}$ denotes the instruction-driven editing pipeline considered in our threat scenario, and ψ denotes the edit instruction.

The central difficulty is to jointly satisfy these requirements under the fixed scene-wide support needed for watermark. Uniformly applying adversarial perturbations over the same support can interfere with watermark recovery, edit deterrence, and rendering fidelity. Moreover, unlike image-domain attacks with direct pixel-space budgets (Salman et al., 2023; Liang et al., 2023; Liang & Wu, 2023; Shan et al., 2023), 3DGS perturbations are applied to Gaussian parameters with different rendering roles, while fidelity is observed only after rendering. Thus, our framework must control both the spatial support and parameter-role sensitivity of adversarial updates.

4 Method

4.1 Overview

Figure 2 illustrates the proposed framework. GuardMarkGS consists of two optimization branches: a scene-wide watermarking branch for ownership tracing, described in Section 4.2, and an adversarial edit-deterrence branch, described in Section 4.3. At iteration t, we sample a training view set $\mathcal{V}_t$, which is shared by both branches. The watermarking branch optimizes all rendered views in $\mathcal{V}_t$ to embed a recoverable ownership message into the protected 3DGS scene. For the adversarial branch, we precompute a fixed soft Gaussian mask from text-guided multi-view segmentation before optimization. This mask controls the adversarial update strength of each Gaussian, assigning stronger edit-deterrence updates to mask-selected Gaussians while allowing surrounding Gaussians to receive weaker updates. We further modulate adversarial gradients according to the rendering role of each Gaussian parameter, mitigating the mismatch between Gaussian-parameter perturbations and



(a) *bear* (b) *face*

Figure 3: **Visualization of adversarially updated Gaussians.** Without the proposed Gaussian selection, the adversarial objective naturally forms regions of large adversarial updates around the main scene objects in easier-to-defend scenes such as *bear*. However, in more difficult-to-defend scenes such as *face*, these update patterns become spatially scattered. Based on this observation, we use update-saliency-motivated selection to select Gaussians where defense-effective updates are concentrated and assign them stronger adversarial updates.

rendered-image fidelity. We optimize the protected scene with the following overall objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{wm}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \tag{4}$$

where $\mathcal{L}_{\text{wm}}$ denotes the scene-wide watermarking objective for ownership tracing, $\mathcal{L}_{\text{adv}}$ denotes the adversarial objective for edit deterrence, and λ_{adv} controls the relative strength of the edit-deterrence branch. The watermarking objective is applied scene-wide, whereas adversarial gradients are scaled by soft mask weights and parameter-role coefficients, enabling stronger updates on mask-selected Gaussians and more conservative perturbations for visually sensitive parameter groups.

4.2 Scene-wide watermark optimization

The watermark branch follows an iterative message recovery process over protected renders using a pre-trained watermark decoder (Zhu et al., 2018). Since low-frequency components are more robust to common distortions, we decode the watermark from the low-frequency wavelet band of each rendered view (Lin & Chen, 2000). We define the watermark objective as follows:

$$\mathcal{L}_{\text{wm}} = \lambda_{\text{msg}} \mathcal{L}_{\text{msg}} + \lambda_{\text{quality}} \mathcal{L}_{\text{quality}}, \tag{5}$$

where λ_{msg} and λ_{quality} control the relative weights of message recovery and rendering fidelity. Both $\mathcal{L}_{\text{msg}}$ and $\mathcal{L}_{\text{quality}}$ are computed for each rendered view in $\mathcal{V}_t$. Here, $\mathcal{L}_{\text{msg}}$ is binary cross entropy between decoded bits and the target message, and $\mathcal{L}_{\text{quality}}$ combines an ℓ_1 term and an LPIPS (Zhang et al., 2018) term between protected renders and corresponding frozen reference renders. We optimize this watermarking objective scene-wide because the watermark signal should remain recoverable from diverse viewpoints and robust to realistic image- or model-level distortions.

4.3 Adversarial edit deterrence

4.3.1 Update-saliency-motivated Gaussian selection

Figure 3 visualizes the Gaussians whose parameters are strongly updated by the adversarial terms later defined in Equation (8). Without the proposed Gaussian selection, easier-to-defend scenes such as *bear* naturally show concentrated update-salient patterns around the main scene objects, whereas more difficult-to-defend scenes such as *face* exhibit irregular and spatially scattered updates. This observation suggests that effective edit deterrence benefits from concentrating the adversarial budget on a selected Gaussian subset rather than perturbing the entire scene uniformly. Motivated by this observation, we precompute text-guided 2D semantic masks (Kirillov et al., 2023) before optimization and convert them into per-Gaussian mask scores by accumulating the mask values weighted by each Gaussian's rasterized pixel contribution. For each Gaussian g_i, we compute

$$s_i = \frac{\sum_{v \in \mathcal{V}_{\text{mask}}} \sum_p a_{i,v}(p)\, M_v(p)}{\sum_{v \in \mathcal{V}_{\text{mask}}} \sum_p a_{i,v}(p) + \epsilon}, \tag{6}$$

where M_v is the 2D mask at view v and $a_{i,v}(p)$ denotes the rasterized contribution of Gaussian g_i to pixel p. Instead of using a strict binary mask, we convert the score into a soft per-Gaussian

coefficient that controls the strength of adversarial updates:

$$m_i = \min\left(1, \left(\frac{s_i}{\tau_{\text{mask}} + \epsilon}\right)^{\gamma_{\text{mask}}}\right). \tag{7}$$

Here, τ_{mask} controls the score level at which the adversarial update scale saturates, while γ_{mask} controls how sharply this update coefficient decays for Gaussians with lower mask scores. In practice, $\mathcal{V}_{\text{mask}}$ is sampled once before optimization, and the soft mask is kept fixed. The adversarial gradient for each Gaussian is scaled by m_i, so that mask-selected Gaussians receive stronger updates while surrounding Gaussians receive weaker updates. Rather than enforcing a hard separation between selected and surrounding Gaussians, the soft coefficient applies stronger adversarial updates to selected Gaussians while allowing weaker updates on nearby Gaussians. This soft allocation helps preserve rendering fidelity and watermark recovery by avoiding overly localized or overly aggressive perturbations. Experiments comparing hard and soft masking appear in Appendix J.

4.3.2 Adversarial objective

The adversarial branch aims to make the editing trajectory diverge from the intended one. For the sampled view set $\mathcal{V}_t$, we optimize an averaged multi-view objective:

$$\mathcal{L}_{\text{adv}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \left(\mathcal{L}^{(v)}_{\text{render}} - \lambda_{\text{lat}}\, \mathcal{S}^{(v)}_{\text{lat}} - \lambda_{\text{traj}}\, \mathcal{S}^{(v)}_{\text{traj}} - \lambda_{\text{xattn}}\, \mathcal{S}^{(v)}_{\text{xattn}}\right). \tag{8}$$

Here, $\mathcal{L}^{(v)}_{\text{render}}$ measures reconstruction error against the corresponding frozen reference render, λ_{lat}, λ_{traj}, and λ_{xattn} control the relative contributions of the three diversion terms, $\mathcal{G}^{\text{ref}}$ denotes the frozen pretrained 3DGS scene, and $z(\cdot)$ is the VAE encoder. The latent-anchor separation term increases separation in the VAE latent space between the protected and frozen reference renders:

$$\mathcal{S}^{(v)}_{\text{lat}} = \left\| z\big(\mathcal{R}_v(\mathcal{G}^{\text{prot}})\big) - z\big(\mathcal{R}_v(\mathcal{G}^{\text{ref}})\big) \right\|_2^2. \tag{9}$$

The denoising-trajectory diversion term, $\mathcal{S}^{(v)}_{\text{traj}}$, separates the protected render from the frozen reference render under the same edit prompt ψ, diffusion timestep t, and sampled noise ϵ (Mokady et al., 2023). Let $\Phi_{\text{traj}}(I; \psi, t, \epsilon)$ denote the frozen editor's denoising-trajectory descriptor summarizing the prompt-induced edit direction and short denoising trajectory from image I. We define

$$\mathcal{S}^{(v)}_{\text{traj}} = \left\| \Phi_{\text{traj}}\big(\mathcal{R}_v(\mathcal{G}^{\text{prot}}); \psi, t, \epsilon\big) - \Phi_{\text{traj}}\big(\mathcal{R}_v(\mathcal{G}^{\text{ref}}); \psi, t, \epsilon\big) \right\|_2^2. \tag{10}$$

This term pushes the protected render away from the reference editing direction even when the protected and reference renders remain visually similar before editing.

We additionally use a cross-attention diversion term, $\mathcal{S}^{(v)}_{\text{xattn}}$, which enlarges the query separation in the mid-block text-conditioned cross-attention of the frozen diffusion editor (Hertz et al., 2023):

$$\mathcal{S}^{(v)}_{\text{xattn}} = \frac{1}{d_{\text{mid}}} \left\| \Phi^{\text{xattn}}_{\text{mid}}\big(\mathcal{R}_v(\mathcal{G}^{\text{prot}}); \psi, t, \epsilon\big) - \Phi^{\text{xattn}}_{\text{mid}}\big(\mathcal{R}_v(\mathcal{G}^{\text{ref}}); \psi, t, \epsilon\big) \right\|_2^2, \tag{11}$$

where $\Phi^{\text{xattn}}_{\text{mid}}$ denotes a descriptor obtained by pooling query tokens from mid-block text-conditioned cross-attention. By increasing this diversion term, the protected render induces prompt-to-image alignment that differs from that of the reference, disrupting the intended editing trajectory. For a detailed definition of the cross-attention diversion term in Equation (11), refer to Appendix I.

4.3.3 Parameter-role-aware adversarial modulation

The update-saliency-motivated Gaussian selection determines which Gaussians should receive stronger adversarial signals. However, it does not resolve another source of imbalance: each Gaussian consists of parameters with different rendering roles whose perturbations have different effects on rendered images. In particular, the scale s_i and rotation r_i determine the anisotropic covariance of a Gaussian, opacity α_i controls visibility and alpha compositing, the DC color coefficient $\mathbf{c}^{\text{dc}}_i$ controls view-independent appearance, and the remaining spherical-harmonic coefficients $\mathbf{c}^{\text{rest}}_i$ control view-dependent residual appearance. Therefore, applying the same adversarial gradient scale to all parameter groups does not reliably control the resulting rendered-image degradation.

Table 1: **Quantitative results.** Results are averaged over the benchmark. PSNR, SSIM, LPIPS, and bit accuracy are measured on renders. CLIP, CLIP-T, and CLIP-D are measured after editing with DGE (Chen et al., 2024a). 3DGSW+DEGauss denotes a straightforward composition of watermarking and adversarial objectives. Orig. denotes values after editing a pretrained 3DGS. GuardMarkGS achieves the highest sUCPS, jointly evaluating traceability, edit deterrence, and fidelity.

Method	sUCPS ↑	Bit Acc. (%) ↑	CLIP			CLIP-T			CLIP-D			PSNR ↑	SSIM ↑	LPIPS ↓
			Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑			
3DGSW (Jang et al., 2025)	0.7791	**99.00**	1.0000	0.9629	0.0371	0.2825	0.2826	-0.0001	0.1476	0.1408	0.0068	33.94	0.9505	0.0866
GaussianMarker (Huang et al., 2024c)	0.7516	98.51		0.9731	0.0269		0.2835	-0.0010		0.1537	-0.0061	**35.37**	**0.9710**	**0.0597**
GuardSplat (Chen et al., 2025)	0.7489	98.92		0.9854	0.0146		0.2819	0.0006		0.1492	-0.0016	29.34	0.9225	0.0627
DEGauss (Meng et al., 2025)	0.6467	N/A		0.9250	0.0750		0.2713	0.0112		0.1268	0.0208	30.34	0.9120	0.1502
3DGSW+DEGauss	0.6200	62.79		0.9596	0.0404		0.2813	0.0012		0.1387	0.0089	29.90	0.8463	0.1950
GuardMarkGS (Ours)	**0.8622**	97.23		**0.9093**	**0.0907**		**0.2676**	**0.0149**		**0.1086**	**0.0390**	30.36	0.8935	0.1471

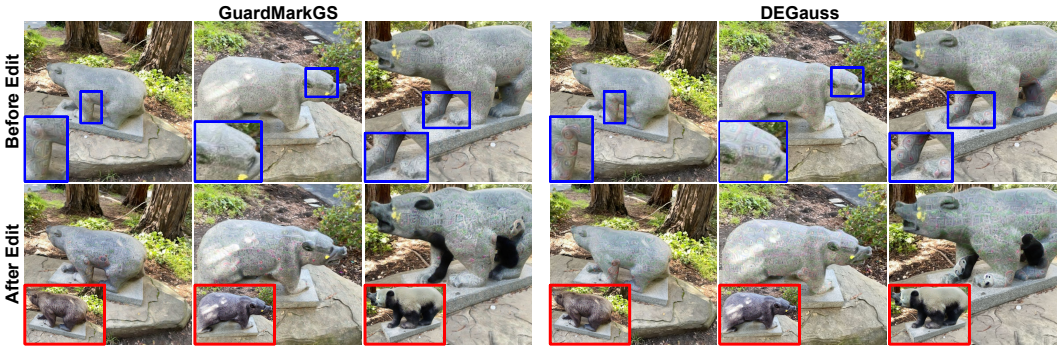


Figure 4: **Qualitative results.** We compare pre- and post-edit renders using DGE (Chen et al., 2024a), with DEGauss (Meng et al., 2025) as the editing-defense baseline. The two rows show pre- and post-edit results, respectively. *Blue boxes* highlight regions for assessing pre-edit rendering fidelity, while *red boxes* indicate edited regions of the pretrained 3DGS. Our method provides both traceability and edit deterrence while preserving higher rendering fidelity than DEGauss.

To account for the different rendering sensitivities of Gaussian parameter groups, we assign a parameter-role coefficient ρ_k to parameter group k and modulate only the adversarial gradient:

$$\widetilde{\nabla}_{\theta_{i,k}} \mathcal{L}_{\text{adv}} = m_i \, \rho_k \, \nabla_{\theta_{i,k}} \mathcal{L}_{\text{adv}}, \tag{12}$$

where $\theta_{i,k}$ denotes the k-th parameter group of Gaussian g_i. The final gradient $\nabla_{\theta_{i,k}} \mathcal{L}$ used to update the protected scene combines the watermark gradient with the modulated adversarial gradient:

$$\nabla_{\theta_{i,k}} \mathcal{L} = \nabla_{\theta_{i,k}} \mathcal{L}_{\text{wm}} + \lambda_{\text{adv}} \, \widetilde{\nabla}_{\theta_{i,k}} \mathcal{L}_{\text{adv}}. \tag{13}$$

The parameter-role coefficients are chosen to reflect the rendering sensitivity of each parameter role: opacity and DC color are treated more conservatively because they can cause visible density or color drift across views, whereas scale, rotation, and view-dependent residual features can carry adversarial signals with less direct global appearance shift.

5 Experiments

5.1 Experimental setup

Scenes and datasets. We evaluate on *bicycle* and *garden* from Mip-NeRF 360 (Barron et al., 2022), and *face*, *person*, *bear*, and *fangzhou* from Instruct-NeRF2NeRF (Haque et al., 2023). All evaluations are conducted on novel views, so that our method and the baselines are compared fairly under the same rendering setting.

Baselines. We compare against watermarking baselines, 3DGSW (Jang et al., 2025), GaussianMarker (Huang et al., 2024c), and GuardSplat (Chen et al., 2025), as well as one defense-only baseline, DEGauss (Meng et al., 2025). To examine whether joint protection can be obtained by a simple composition of existing methods, we include 3DGSW+DEGauss, which adds DEGauss-style adversarial terms to a 3DGSW-style watermarking objective.

Editing pipelines. We use DGE (Chen et al., 2024a) as the main editing pipeline, a representative 3DGS editing method enforcing multi-view consistency. We evaluate defense performance using GaussianEditor (Chen et al., 2024b) to examine whether edit deterrence transfers to another pipeline.

Table 2: **Ablation study of the joint optimization design.** Since watermarking and adversarial objectives are essential for joint optimization, we omit them from the columns and present ablation studies on update-saliency-motivated Gaussian selection and parameter-role-aware adversarial modulation. Results show that the proposed components improve the trade-off among traceability, edit deterrence, and rendering fidelity, with the full model achieving the best balance.

Gaussian Selection	Adversarial Modulation	sUCPS ↑	Bit Acc. (%) ↑	CLIP			CLIP-T			CLIP-D			PSNR ↑	SSIM ↑	LPIPS ↓
				Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑			
-	-	0.6776	67.80	1.0000	**0.8706**	**0.1294**	0.2825	**0.2620**	**0.0205**	0.1476	**0.0851**	**0.0625**	28.73	0.8159	0.3069
✓	-	0.8566	97.02		0.9088	0.0912		0.2678	0.0147		0.1098	0.0378	30.22	0.8801	0.1503
-	✓	0.6683	65.02		0.8734	0.1266		0.2625	0.0200		0.0890	0.0586	28.74	0.8161	0.3062
✓	✓	**0.8622**	**97.23**		0.9093	0.0907		0.2676	0.0149		0.1086	0.0390	**30.36**	**0.8935**	**0.1471**

Evaluation metrics. We evaluate traceability using watermark bit accuracy, edit deterrence using CLIP (Radford et al., 2021), CLIP-T, and CLIP-D after 3D editing, and rendering fidelity using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). We additionally define the *soft Unified Copyright Protection Score* (sUCPS) to summarize the balance among traceability, edit deterrence, and fidelity. Detailed metric definitions are provided in Appendices G and H.

Implementation details. Implementation settings and edit prompts are provided in Appendix C.

5.2 Main Results

Quantitative results. Table 1 summarizes the averaged quantitative results over the benchmark set, including the proposed unified score sUCPS. The watermark-only methods achieve high bit accuracy and strong rendering fidelity, but they show clear limitations on the edit-deterrence metrics. DEGauss is included as a defense-only reference, but it does not provide watermark decoding and therefore cannot satisfy the ownership-tracing requirement. The 3DGSW+DEGauss further tests a straightforward composition of existing watermarking and edit-deterrence objectives. This naive combination fails to achieve a stable balance, as it substantially reduces bit accuracy and rendering quality while providing only limited edit-deterrence gains. These results show that joint protection cannot be obtained by simply adding adversarial edit-deterrence terms to a watermarking objective. In contrast, our method demonstrates recoverable ownership tracing with high bit accuracy and effective editing defense while substantially mitigating the degradation in watermark recovery and rendering fidelity observed under naive joint optimization. Taken together, these results suggest that GuardMarkGS provides a more stable balance among ownership tracing, edit deterrence, and rendering fidelity than a straightforward composition of the two protection objectives.

Qualitative results. Figure 4 compares our method with DEGauss (Meng et al., 2025) before and after 3D editing using DGE (Chen et al., 2024a). The defense-only baseline introduces visible degradation in the pre-edit renders, whereas our method maintains higher rendering fidelity while still disrupting the edited results. These qualitative results indicate that our method provides a better balance between fidelity and edit deterrence, in addition to supporting ownership traceability through watermark recovery. Additional qualitative results are provided in Appendix M.

Additional generalization experiments. We evaluate edit deterrence under two complementary settings: GaussianEditor (Chen et al., 2024b), an alternative 3D editing pipeline, for cross-editor transfer, and DGE (Chen et al., 2024a) with a different image-editing diffusion backbone for cross-backbone transfer. Results are provided in Appendices D and E, respectively.

5.3 Ablation Study

Effect of update-saliency-motivated selection. Table 2 reports the ablation of update-saliency-motivated Gaussian selection under the same scene-wide watermarking and adversarial objectives. When parameter-role modulation is enabled, adding the proposed selection strategy improves bit accuracy from 65.02 to 97.23, PSNR from 28.74 to 30.36, and SSIM from 0.8161 to 0.8935, while reducing LPIPS from 0.3062 to 0.1471. Although the variants without selection can produce stronger raw CLIP-based edit-deterrence gaps, they do so with severe degradation in watermark recovery and rendering fidelity. This indicates that Gaussian selection mainly stabilizes the joint optimization. Additional results comparing hard and soft Gaussian masking are provided in Appendix J.

Effect of parameter-role modulation. Table 2 also assesses the effect of the role-aware coefficients in Equation (12). Fixing the proposed Gaussian selection, adding parameter-role-aware modulation

Table 3: **Ablation on adversarial diversion-term combinations.** We evaluate combinations of latent-anchor separation $\mathcal{S}_{\text{lat}}$, denoising-trajectory diversion $\mathcal{S}_{\text{traj}}$, and cross-attention diversion $\mathcal{S}_{\text{xattn}}$. Results show that the three terms work complementarily to achieve strong edit deterrence.

$\mathcal{S}_{\text{lat}}$	$\mathcal{S}_{\text{traj}}$	$\mathcal{S}_{\text{xattn}}$	sUCPS ↑	Bit Acc. (%) ↑	CLIP			CLIP-T			CLIP-D			PSNR ↑	SSIM ↑	LPIPS ↓
					Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑			
-	✓	✓	0.8621	95.35		0.9251	0.0749		0.2712	0.0113		0.1232	0.0244	30.02	0.8956	0.1398
✓	-	✓	0.7733	97.12		0.9562	0.0438		0.2795	0.0030		0.1426	0.0050	31.24	0.9022	0.1301
✓	✓	-	0.8595	95.42		0.9224	0.0776		0.2737	0.0088		0.1237	0.0239	30.32	0.8944	0.1333
-	-	✓	0.7686	96.59	1.0000	0.9642	0.0358	0.2825	0.2798	0.0027	0.1476	0.1412	0.0064	**31.28**	0.9043	0.1298
✓	-	-	0.7325	97.12		0.9644	0.0356		0.2814	0.0011		0.1474	0.0002	31.20	**0.9067**	0.1288
-	✓	-	0.8402	95.13		0.9306	0.0694		0.2747	0.0078		0.1334	0.0142	30.22	0.8999	**0.1286**
✓	✓	✓	**0.8622**	**97.23**		**0.9093**	**0.0907**		**0.2676**	**0.0149**		**0.1086**	**0.0390**	30.36	0.8935	0.1471

improves sUCPS from 0.8566 to 0.8622, bit accuracy from 97.02 to 97.23, and PSNR from 30.22 to 30.36. The CLIP-based edit-deterrence scores remain comparable, with CLIP changing from 0.9088 to 0.9093, CLIP-T from 0.2678 to 0.2676, and CLIP-D from 0.1098 to 0.1086. These results suggest that parameter-role-aware modulation further improves the fidelity side of the trade-off while preserving the edit-deterrence effect of the selected adversarial updates. Additional ablations on the perturbed Gaussian parameter groups are provided in Appendix K.

Diversion-term analysis. We analyze how the three diversion terms in Equation (8) contribute to editing defense performance. Specifically, we examine latent-anchor separation $\mathcal{S}_{\text{lat}}$ in Equation (9), denoising-trajectory diversion $\mathcal{S}_{\text{traj}}$ in Equation (10), and cross-attention diversion $\mathcal{S}_{\text{xattn}}$ in Equation (11). Table 3 shows that rendering fidelity and bit accuracy remain comparable across variants, whereas edit-deterrence metrics vary more substantially. This indicates that the performance gain does not simply come from injecting more adversarial noise, but from the complementary effects of the three diversion terms. Using all three terms achieves stronger editing defense, resulting in a better balance across traceability, edit deterrence, and rendering fidelity, yielding the highest sUCPS.

Ablation of the joint protection design. Table 2 summarizes how the two allocation mechanisms interact under the fixed watermarking and adversarial objectives. Without both Gaussian selection and role-aware modulation, the model produces strong raw CLIP-based edit-deterrence gaps, but suffers from low bit accuracy and poor rendering fidelity, indicating a severe conflict between the two protection signals. Gaussian selection provides the main improvement in watermark recovery and rendering fidelity by controlling the spatial support of adversarial updates. Role-aware modulation further refines this allocation by scaling adversarial gradients according to Gaussian parameter roles. The full model achieves the best overall sUCPS, suggesting that effective joint protection requires explicit control over both the selected Gaussian support and the parameter-role sensitivity, rather than simply applying watermarking and adversarial objectives to the same 3DGS support.

6 Conclusion

We present a unified protection framework for 3D Gaussian Splatting that jointly addresses ownership tracing and unauthorized edit deterrence. The core idea is to embed the watermark signal over the full scene while assigning stronger adversarial updates to a mask-selected Gaussian subset. This design addresses a central tension in 3DGS copyright protection: ownership evidence should remain recoverable across diverse viewpoints through scene-wide watermarking, while adversarial signals should be selectively allocated under joint optimization rather than uniformly imposed on the same scene-wide support. This allocation helps maintain a stable balance among traceability, edit deterrence, and rendering fidelity. The experimental results show that the proposed framework achieves the best sUCPS, our unified score that jointly reflects traceability, edit deterrence, and fidelity, while maintaining competitive performance across all three aspects. We believe that future 3D copyright protection systems should move beyond isolated objectives, such as passive ownership tracing or active edit deterrence, and instead optimize them jointly.

Limitations and future work. We have shown that copyright tracing via digital watermarking and active copyright protection through adversarial signal injection can be achieved simultaneously within a unified framework. However, to become a practical copyright protection technique, the method must demonstrate robustness under a wider range of distortion scenarios, including not only distortions applied to rendered images but also model-level perturbations that directly modify the protected 3D representation. Therefore, future work should include a broader and more systematic analysis under realistic copyright-protection scenarios that incorporate such distortions.

AI USE STATEMENT

We used AI in a limited manner for grammar and style polishing only. It did not contribute to research ideation, experimental design, and analysis. The authors take full responsibility for all content.

REPRODUCIBILITY STATEMENT

We provide implementation and evaluation details in Section 5. Appendix C further describes the optimization procedure in Algorithm 1 and provides the evaluation prompts. Additional evaluation and ablation details are provided in the subsequent appendices.

REFERENCES

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

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402, June 2023.

Minghao Chen, Iro Laina, and Andrea Vedaldi. DGE: direct gaussian 3d editing by consistent multi-view editing. In *ECCV (74)*, pp. 74–92, 2024a.

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and Controllable 3D Editing with Gaussian Splatting . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21476–21485, June 2024b.

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 (CVPR)*, pp. 16325–16335, June 2025.

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 (ICCV)*, pp. 22466–22477, October 2023.

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

Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19740–19750, October 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *International Conference on Learning Representations (ICLR)*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24. Association for Computing Machinery, 2024a. ISBN 9798400705250. doi: 10.1145/3641519.3657428.

Huayang Huang, Yu Wu, and Qian Wang. Robin: Robust and invisible watermarks for diffusion models with adversarial optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 3937–3963. Curran Associates, Inc., 2024b. doi: 10.52202/079017-0129.

Xiufeng Huang, Ruiqi Li, Yiu ming Cheung, Ka Chun Cheung, Simon See, and Renjie Wan. Gaussianmarker: Uncertainty-aware copyright protection of 3d gaussian splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c.

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*, pp. 12087–12097, 2024.

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 (CVPR)*, pp. 5938–5948, June 2025.

Joonsung Jeon, Woo Jae Kim, Suhyeon Ha, Sooel Son, and Sung eui Yoon. Advpaint: Protecting images from inpainting manipulation via adversarial attention disruption. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jaehwan Jeong, Sumin In, Sieun Kim, Hannie Shin, Jongheon Jeong, Sang Ho Yoon, Jaewook Chung, and Sangpil Kim. Faceshield: Defending facial image against deepfake threats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10364–10374, October 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Sangheon Shin, Sangmin Kim, and Sangpil Kim. Editsplat: Multi-view fusion and attention-guided optimization for view-consistent 3d scene editing with 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 11135–11145, June 2025.

Chenxin Li, Brandon Y. Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang. Steganerf: Embedding invisible information within neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 441–453, October 2023.

Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023.

Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: preventing painting imitation from diffusion models via adversarial examples. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

S. D. Lin and Chin-Feng Chen. A robust dct-based watermarking for copyright protection. *IEEE Trans. on Consum. Electron.*, 46(3):415–421, August 2000. ISSN 0098-3063. doi: 10.1109/30.883387.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2024.

Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22401–22411, October 2023.

Lingzhuang Meng, Mingwen Shao, Yuanjian Qiao, and Xiang Lv. DEGauss: Defending against malicious 3d editing for gaussian splatting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 405–421. Springer International Publishing, 2020.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047, June 2023.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), July 2022. doi: 10.1145/3528223.3530127.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/081b08068e4733ae3e7ad019fe8d172f-Paper-Conference.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, June 2022.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-e: Text-guided voxel editing of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 430–440, October 2023.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by Text-to-Image models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2187–2204, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 33879–33896, 2024.

Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N. Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2116–2127, October 2023.

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3835–3844, June 2022.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 58047–58063. Curran Associates, Inc., 2023.

Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian D. Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *ECCV (14)*, pp. 55–71, 2024.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6796–6807, June 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 (CVPR)*, pp. 19447–19456, June 2024a.

Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Trans. Graph.*, 43(6), November 2024b. ISSN 0730-0301. doi: 10.1145/3687937.

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 (CVPR)*, June 2018.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pp. 682–697. Springer-Verlag, 2018. ISBN 978-3-030-01266-3. doi: 10.1007/978-3-030-01267-0_40.

Jingyu Zhuang, Chen Wang, Liang Lin, Lingjie Liu, and Guanbin Li. Dreameditor: Text-driven 3d scene editing with neural fields. In *SIGGRAPH Asia 2023 Conference Papers*. Association for Computing Machinery, 2023. doi: 10.1145/3610548.3618190.

A BROADER IMPACT

The proposed framework can help creators and organizations protect 3D assets in digital heritage, virtual production, AR/VR, and simulation pipelines. By combining ownership tracing and edit deterrence within a protected 3DGS representation, our method supports practical copyright protection even when released assets are redistributed or subjected to instruction-driven editing. This can help owners preserve ownership evidence, discourage unauthorized manipulation, and deploy protected 3D assets while maintaining rendering fidelity. More broadly, our work highlights the need for copyright protection systems that jointly consider traceability, edit deterrence, and visual quality, providing a foundation for future research on reliable protection of editable 3D representations.

B THREAT MODEL AND APPLICATION SCENARIO

We consider the post-release setting illustrated in Figure 1. Alice starts from a pretrained 3DGS scene, optimizes it with GuardMarkGS, and then releases or distributes the protected representation. An unauthorized user, Bob, may obtain the released 3DGS, render it from chosen viewpoints, and use it as input to an instruction-driven 3DGS editing pipeline with a target instruction ψ. GuardMarkGS is applied before release, whereas ownership verification and edit-deterrence evaluation are performed after release.

Ownership verification. Alice retains the predefined ownership message $\mathbf{m}$ and the watermark decoder $\mathcal{D}_{\text{msg}}$. Given a protected or disputed 3DGS, Alice renders novel views and decodes the ownership message from the rendered images. We measure tracing performance by the resulting bit accuracy, consistent with the evaluation protocol in Section 5.1.

Unauthorized editing. For evaluation, we apply the same target instruction to the original and protected scenes through the evaluated editing pipeline. The protection is considered more effective when the edited protected result follows the requested semantic transformation less strongly than the corresponding edit of the original scene, while the pre-edit protected rendering remains close to the original rendering. This behavior is quantified using the CLIP-based edit-deterrence metrics and rendering-fidelity metrics described in Section 5.1 and Appendix G.

Robustness setting. In addition to direct instruction-driven editing, we separately evaluate image- and model-level distortions that may weaken the embedded protection signals. These experiments are reported in Appendix F.

C ADDITIONAL ALGORITHMIC DETAILS

In this section, we provide implementation and algorithmic details for the experiments in Section 5 of the main paper.

Experiments use an NVIDIA RTX A6000. Each scene is optimized for 8 epochs using a 32-bit watermark message, and target Gaussians for adversarial noise injection are selected using 2D masks from 24 viewpoints. We set $\lambda_{\text{msg}} = 0.1$, $\lambda_{\text{quality}} = 1.0$, and $\lambda_{\text{lat}} = \lambda_{\text{traj}} = \lambda_{\text{xattn}} = 10^{-4}$.

The overall optimization procedure is summarized in Algorithm 1. Table 4 lists the edit text prompts used to evaluate edit-deterrence performance, including the main evaluation reported in Table 1.

D ADDITIONAL RESULTS UNDER DIFFERENT EDITING PIPELINE

We provide additional quantitative results using GaussianEditor (Chen et al., 2024b), a 3D editing pipeline other than DGE (Chen et al., 2024a). Table 5 reports edit-deterrence results when GaussianEditor is used. In addition, Figure 5 shows qualitative results edited with GaussianEditor. Taken together, these results show that the edit-deterrence performance of our proposed method is not limited to a specific editing model.

Algorithm 1 Joint optimization of GuardMarkGS

Require: Pretrained 3DGS scene $\mathcal{G}$, frozen reference scene $\mathcal{G}^{\text{ref}} \leftarrow \mathcal{G}$, training view set $\mathcal{V}$, ownership message $\mathbf{m}$, prompt library Ψ_{lib}, parameter-role coefficients $\{\rho_k\}$, total iterations T
Ensure: Protected 3DGS scene $\mathcal{G}^{\text{prot}}$

Initialize $\mathcal{G}^{\text{prot}} \leftarrow \mathcal{G}$
Precompute a soft Gaussian mask $\{m_i\}$ from text-guided multi-view semantic segmentation
Set role coefficients $\{\rho_k\}$ for Gaussian parameter groups
for $t = 1$ to T **do**
  Sample a multi-view set $\mathcal{V}_t \subset \mathcal{V}$
  Sample an edit prompt $\psi \sim \Psi_{\text{lib}}$
  for each $v \in \mathcal{V}_t$ **do**
    Render protected view $I_v^{\text{prot}} = \mathcal{R}_v(\mathcal{G}^{\text{prot}})$
    Render reference view $I_v^{\text{ref}} = \mathcal{R}_v(\mathcal{G}^{\text{ref}})$
    Compute watermark loss $\mathcal{L}_{\text{wm}}^{(v)}$
    Compute adversarial loss $\mathcal{L}_{\text{adv}}^{(v)}$
  end for
  Aggregate the scene-wide watermark loss over $\mathcal{V}_t$

$$\mathcal{L}_{\text{wm}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \mathcal{L}_{\text{wm}}^{(v)}$$

  Aggregate the averaged adversarial loss over $\mathcal{V}_t$

$$\mathcal{L}_{\text{adv}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \mathcal{L}_{\text{adv}}^{(v)}$$

  Backpropagate $\mathcal{L}_{\text{wm}}$ to obtain scene-wide watermark gradients
  Compute adversarial gradients $\nabla_{\theta_{i,k}} \mathcal{L}_{\text{adv}}$
  Apply the soft Gaussian mask and role coefficients:

$$\widetilde{\nabla}_{\theta_{i,k}} \mathcal{L}_{\text{adv}} = m_i \rho_k \nabla_{\theta_{i,k}} \mathcal{L}_{\text{adv}}$$

  Combine the watermark and λ_{adv}-weighted modulated adversarial gradients
  Update the parameters of $\mathcal{G}^{\text{prot}}$ with the optimizer
end for
return $\mathcal{G}^{\text{prot}}$

E Additional results under a different diffusion backbone

The GaussianEditor experiment in Appendix D changes the 3D editing pipeline. Here, we examine a complementary generalization axis by retaining the multi-view render-edit-update and 3D fitting stages of DGE (Chen et al., 2024a), while replacing its original image editor (Brooks et al., 2023) with an SDXL-based InstructPix2Pix backbone (Podell et al., 2024). This setting tests whether the edit-deterrence effect transfers to another diffusion backbone.

We follow the same evaluation protocol as Table 1, covering six scenes and novel-view renders. The CLIP-based metrics and their differences follow the definitions in Appendix G.

As shown in Table 6, GuardMarkGS maintains stronger edit deterrence than both watermark-only methods and the defense-only DEGauss baseline under the alternative diffusion backbone. These results indicate that its edit-deterrence effect generalizes across diffusion backbones rather than being tied to the specific diffusion backbone.

F Additional robustness experiments

In the main paper, Table 1 reports the primary quantitative comparison results. In this section, we present additional experiments to further examine the robustness of the proposed framework. Here,

Table 4: **Edit prompts used for evaluating edit-deterrence performance.** For each evaluated scene, we list the prompts used for 3D editing under its scene-specific evaluation setting.

scene	edit text prompt	scene	edit text prompt
bear	Turn the bear statue into a panda Turn the bear statue into a grizzly bear Turn the bear statue into an asiatic black bear Turn the bear statue into a wild boar	bicycle	Change the bicycle color to bright red Turn the ground into a Namibian desert A photo of the bicycle at the Namibian desert
face	Wear him a glasses Make him wear a Venetian mask Turn him into the Tolkien Elf Turn him into an Einstein A photo of a marble sculpture	garden	Change the table color to deep mahogany brown Turn the vase into red A photo of a garden scene with autumn
person	Make the man look like a mosaic Sculpture Turn the man into a robot Turn him into a Minecraft character Make him reading a book	fangzhou	Turn him into a comic book character Turn him into an old man with wrinkles Turn him into a spider man with Mask

Table 5: **Generalization to GaussianEditor: quantitative results.** Lower CLIP, CLIP-T, and CLIP-D values indicate more successful editing defense. Here, the Orig. column reports the values measured from edits of the original scene, while the Method column reports the values measured from edits of the protected scene produced by each method. GuardMarkGS also achieves the strongest editing defense performance in experiments using GaussianEditor (Chen et al., 2024b).

Method	CLIP			CLIP-T			CLIP-D		
	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑
3DGSW (Jang et al., 2025)		0.9582	0.0418		0.2770	-0.0033		0.1306	-0.0013
GaussianMarker (Huang et al., 2024c)		0.9643	0.0357		0.2779	-0.0042		0.1449	-0.0156
GuardSplat (Chen et al., 2025)	1.0000	0.9777	0.0223	0.2737	0.2754	-0.0017	0.1293	0.1362	-0.0069
DEGauss (Meng et al., 2025)		<u>0.9274</u>	<u>0.0726</u>		<u>0.2711</u>	<u>0.0026</u>		<u>0.1282</u>	<u>0.0011</u>
GuardMarkGS (Ours)		**0.9243**	**0.0757**		**0.2696**	**0.0041**		**0.1196**	**0.0097**

robustness refers to the ability to preserve both ownership tracing and defense against 3D editing even under scenarios where copyright-protection mechanisms may be weakened.

Robustness of the watermark. In the application scenario of Figure 1, Alice verifies ownership by recovering the embedded watermark from rendered images of the protected 3DGS model. However, an unauthorized user may attempt to weaken or remove the watermark by applying distortions either to the rendered images or directly to the 3DGS model itself. Table 7 reports the watermark recovery results under these image- and model-level distortion scenarios. The results show that the proposed method preserves competitive watermark recovery performance even after such distortions, despite jointly optimizing watermark and adversarial signals.

Robustness of adversarial noise. In the application scenario of Figure 1, Bob may attempt unauthorized editing using the acquired 3DGS model. Furthermore, if Bob is aware that adversarial optimization has been applied to deter such editing, he may try to weaken the protection by adding random noise to the Gaussians that compose the scene, or by performing operations such as random Gaussian cloning or random Gaussian pruning. Table 8 reports the results under these model-distortion scenarios and evaluates how successfully the proposed method continues to deter editing. The results suggest that the proposed framework preserves its edit-deterrence capability even under such distortions.

G Evaluation Metrics

Traceability. We measure ownership traceability using watermark bit accuracy. For each protected 3DGS, we render images from novel viewpoints, decode the ownership message from the rendered images, and compare the decoded bits with the target ownership message.

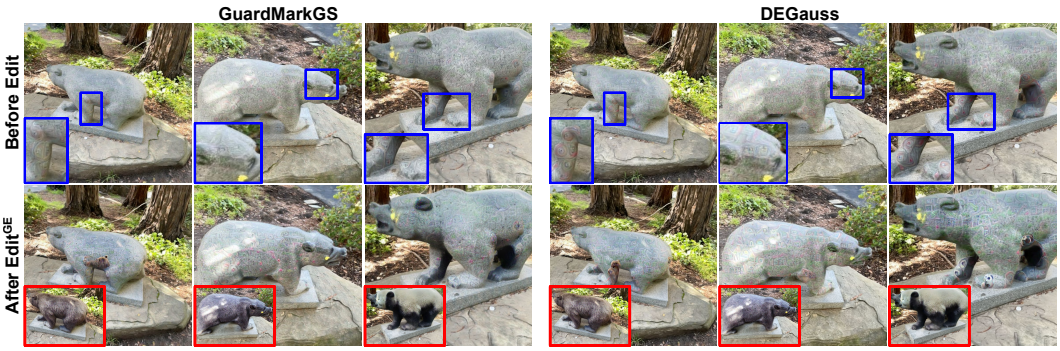


Figure 5: **Generalization to GaussianEditor: qualitative results.** We evaluate our method on GaussianEditor (Chen et al., 2024b), an alternative 3D editing method to DGE (Chen et al., 2024a). *After Edit*^{GE} denotes GaussianEditor outputs. *Blue boxes* highlight regions for assessing pre-edit rendering fidelity, while *red boxes* show edited regions of the pretrained 3DGS. These results demonstrate that our edit deterrence generalizes beyond the DGE pipeline.

Table 6: **Generalization to an SDXL-based diffusion backbone: quantitative results.** We retain the DGE (Chen et al., 2024a) multi-view fitting pipeline and replace its original image-editing backbone with an SDXL-based InstructPix2Pix editor (Podell et al., 2024). Orig. and Method report scores from edits of the pretrained and protected 3DGS, respectively. GuardMarkGS achieves the largest difference across all three metrics, demonstrating cross-backbone generalization.

Method	CLIP			CLIP-T			CLIP-D		
	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑	Orig.	Method ↓	Diff. ↑
3DGSW (Jang et al., 2025)		0.9640	0.0360		0.2844	-0.0011		0.1404	-0.0013
GaussianMarker (Huang et al., 2024c)		0.9698	0.0302		<u>0.2816</u>	<u>0.0017</u>		<u>0.1378</u>	<u>0.0013</u>
GuardSplat (Chen et al., 2025)	1.0000	0.9806	0.0194	0.2833	0.2819	0.0014	0.1391	0.1400	-0.0009
DEGauss (Meng et al., 2025)		<u>0.9478</u>	<u>0.0522</u>		0.2826	0.0007		0.1399	-0.0008
GuardMarkGS (Ours)		**0.9424**	**0.0576**		**0.2809**	**0.0024**		**0.1289**	**0.0102**

Edit deterrence. We use three CLIP-based metrics to quantify edit deterrence: CLIP, CLIP-T, and CLIP-D. All metrics are evaluated on the same novel-view set after applying the editing pipeline. In Table 1 of the main paper, the *Orig.* columns denote scores measured from edits of the original pretrained 3DGS scene, the *Method* columns denote scores measured from edits of the protected scene produced by each method, and *Diff.* denotes the gap between them. Effective edit deterrence does not require the post-edit output to become visually invalid. A visually plausible output can still indicate successful deterrence when the requested semantic transformation is substantially weakened relative to the corresponding edit of the original scene.

Let I_v^{edit} be the edited render of the original scene at view v, and let $I_{i,v}^{\text{edit}}$ be the edited render of the protected scene produced by method i. We use CLIP image and text encoders, denoted by $f_I(\cdot)$ and $f_T(\cdot)$, with cosine similarity between normalized embeddings.

CLIP measures the visual similarity between the edited protected result and the edited original result, with $o^{\text{clip}} = 1$ because the edited original result is compared with itself:

$$r_i^{\text{clip}} = \frac{1}{|\mathcal{V}_{\text{eval}}|} \sum_{v \in \mathcal{V}_{\text{eval}}} \cos\big(f_I(I_{i,v}^{\text{edit}}), f_I(I_v^{\text{edit}})\big)\,. \tag{14}$$

CLIP-T measures how well the edited result matches the target edit prompt ψ_{tgt}:

$$o^{\text{clipT}} = \frac{1}{|\mathcal{V}_{\text{eval}}|} \sum_{v} \cos\big(f_I(I_v^{\text{edit}}), f_T(\psi_{\text{tgt}})\big)\,, \qquad r_i^{\text{clipT}} = \frac{1}{|\mathcal{V}_{\text{eval}}|} \sum_{v} \cos\big(f_I(I_{i,v}^{\text{edit}}), f_T(\psi_{\text{tgt}})\big)\,. \tag{15}$$

CLIP-D measures whether the image-editing direction aligns with the text-editing direction. Let I_v^{src} and $I_{i,v}^{\text{src}}$ denote the corresponding pre-edit renders, and let ψ_{src} and ψ_{tgt} be the source and target edit prompts, respectively. We define the text and image directions in CLIP space as

$$d_t = \text{norm}(f_T(\psi_{\text{tgt}}) - f_T(\psi_{\text{src}}))\,, \qquad d_v^{\text{img}} = \text{norm}\big(f_I(I_v^{\text{edit}}) - f_I(I_v^{\text{src}})\big)\,, \tag{16}$$

Table 7: **Robustness of watermark recovery.** Bit acc. (%) is measured in three settings: no distortion, image-level distortions on renders, and model-level distortions on the protected 3DGS.

Method	Bit Acc. (%) ↑									
	none	Model Distortion			Image Distortion					
		noise	prune	clone	noise (σ=0.01)	rotation ($\pm\pi/6$)	scaling (75%)	blur (σ=0.1)	crop (40%)	jpeg (Q=50)
3DGSW (Jang et al., 2025)	**99.00**	**97.46**	**98.69**	**99.00**	**98.92**	**87.02**	**98.84**	**99.00**	85.36	**94.82**
GaussianMarker (Huang et al., 2024c)	98.51	78.27	92.58	95.24	97.51	81.16	96.42	98.51	72.35	73.58
GuardSplat (Chen et al., 2025)	98.92	64.49	80.16	89.26	96.73	81.61	94.95	98.92	**98.58**	67.95
GuardMarkGS (Ours)	97.23	92.94	94.21	97.23	96.73	82.62	96.97	97.23	71.05	87.86

Table 8: **Robustness of edit deterrence.** We apply model-level distortions to each protected 3DGS representation, perform 3D editing, and report CLIP, CLIP-T, and CLIP-D measured on the edited outputs. For each metric group, Orig. denotes the score measured from edits of the pretrained 3DGS, while Method reports scores measured from edits of the protected scene produced by each method.

Method	CLIP ↓					CLIP-T ↓					CLIP-D ↓				
	Orig.	Method				Orig.	Method				Orig.	Method			
		none	noise	prune	clone		none	noise	prune	clone		none	noise	prune	clone
3DGSW (Jang et al., 2025)		0.9629	0.9322	0.9545	0.9602		0.2826	0.2761	0.2808	0.2822		0.1408	0.1241	0.1330	0.1409
GaussianMarker (Huang et al., 2024c)		0.9731	0.9443	0.9665	0.9710		0.2835	0.2774	0.2820	0.2838		0.1537	0.1396	0.1491	0.1554
GuardSplat (Chen et al., 2025)	1.0000	0.9854	0.9521	0.9757	0.9822	0.2825	0.2819	0.2761	0.2815	0.2830	0.1476	0.1492	0.1281	0.1452	0.1499
DEGauss (Meng et al., 2025)		0.9250	0.8965	0.9170	0.9200		0.2713	0.2654	0.2697	0.2706		0.1268	0.1136	0.1233	0.1221
GuardMarkGS (Ours)		**0.9093**	**0.8895**	**0.9081**	**0.9063**		**0.2676**	**0.2631**	**0.2659**	**0.2677**		**0.1086**	**0.1035**	**0.1086**	**0.1092**

$$d_{i,v}^{\text{img}} = \text{norm}\big(f_I(I_{i,v}^{\text{edit}}) - f_I(I_{i,v}^{\text{src}})\big)\,, \tag{17}$$

where $\text{norm}(\cdot)$ denotes ℓ_2 normalization. The CLIP-D scores are then

$$o^{\text{clipD}} = \frac{1}{|\mathcal{V}_{\text{eval}}|}\sum_v \cos(d_v^{\text{img}}, d_t), \qquad r_i^{\text{clipD}} = \frac{1}{|\mathcal{V}_{\text{eval}}|}\sum_v \cos(d_{i,v}^{\text{img}}, d_t). \tag{18}$$

For each metric $k \in \{\text{clip}, \text{clipT}, \text{clipD}\}$, we report the difference

$$\Delta_i^k = o^k - r_i^k. \tag{19}$$

A lower method score r_i^k indicates that the edited protected result is less aligned with the intended edit. Accordingly, a larger positive difference Δ_i^k indicates a larger reduction from the edited original result, and therefore stronger edit deterrence.

Fidelity. We measure rendering fidelity before editing by comparing novel-view renders of each protected 3DGS with the corresponding renders of the original pretrained scene. We report PSNR and SSIM (Wang et al., 2004), for which higher values indicate better fidelity, and LPIPS (Zhang et al., 2018), for which lower values indicate better perceptual similarity.

H Definition of sUCPS

To evaluate the effectiveness of copyright-protection methods, we define the *soft Unified Copyright Protection Score* (sUCPS). This metric jointly reflects practical 3D copyright-protection requirements: ownership traceability, edit deterrence, and rendering fidelity. Rather than evaluating these aspects in isolation, sUCPS rewards methods that maintain balanced performance across all three. The following definitions specify how these aspects are normalized and aggregated into sUCPS.

Let i index a protection method. We define three normalized subscores: a traceability score T_i, an edit-deterrence score E_i, and a fidelity score F_i. All subscores are mapped to $[0, 1]$, where larger values indicate better performance. For compactness, we use $\text{GM}(\cdot)$ and $\text{HM}(\cdot)$ to denote the geometric and harmonic means, respectively, and clip all normalized values to $[0, 1]$.

Traceability score. Let $b_i \in [0, 1]$ be the watermark bit accuracy of method i. For example, a bit accuracy of 97.23% is used as $b_i = 0.9723$ in the sUCPS computation. For methods that do not provide watermark decoding, we assign the chance-level reference $b_i = 0.5$. Let $b_{\max}$ be the best bit accuracy among methods that support watermark decoding. We define

$$T_i = \text{clip}_{[0,1]}\left[\frac{1}{2}\left(1 + \frac{b_i - 0.5}{b_{\max} - 0.5}\right)\right]. \tag{20}$$

This maps chance-level ownership tracing to a neutral reference and assigns higher scores as watermark recovery improves, with bit accuracy approaching the best observed value.

Edit-deterrence score. For each CLIP-based edit-deterrence metric, we measure the gap between the edited original scene and the edited protected scene. Let

$$\Delta_i^{\text{clip}} = o^{\text{clip}} - r_i^{\text{clip}}, \qquad \Delta_i^{\text{clipT}} = o^{\text{clipT}} - r_i^{\text{clipT}}, \qquad \Delta_i^{\text{clipD}} = o^{\text{clipD}} - r_i^{\text{clipD}}, \tag{21}$$

where o denotes the score of the edited original scene and r_i the score of the edited protected result from method i. A larger positive gap indicates stronger suppression of the editing outcome. We normalize each gap, while preserving its sign, by the largest absolute gap among compared methods:

$$u_i^k = \text{clip}_{[0,1]}\left[\frac{1}{2}\left(1 + \frac{\Delta_i^k}{\max_j |\Delta_j^k|}\right)\right], \qquad k \in \{\text{clip}, \text{clipT}, \text{clipD}\}. \tag{22}$$

The edit-deterrence score is then defined as

$$E_i = \text{GM}\left(u_i^{\text{clip}}, u_i^{\text{clipT}}, u_i^{\text{clipD}}\right). \tag{23}$$

Fidelity score. For rendering fidelity, we use PSNR, SSIM, and LPIPS measured on protected novel-view rendered images. Since higher PSNR and SSIM are better while lower LPIPS is better, we define the normalized fidelity scores as

$$p_i = \frac{\text{PSNR}_i}{\max_j \text{PSNR}_j}, \qquad q_i = \frac{\text{SSIM}_i}{\max_j \text{SSIM}_j}, \qquad \ell_i = \frac{\min_j \text{LPIPS}_j}{\text{LPIPS}_i}. \tag{24}$$

We aggregate them using a geometric mean:

$$F_i = \text{GM}(p_i, q_i, \ell_i). \tag{25}$$

Finally, we combine the three subscores using a harmonic mean:

$$\text{sUCPS}_i = \text{HM}(T_i, E_i, F_i). \tag{26}$$

The harmonic mean penalizes methods that perform well in only one or two aspects but fail in the remaining one. Thus, a high sUCPS indicates that a method jointly preserves ownership traceability, edit deterrence, and rendering fidelity.

I Cross-attention diversion

Along with latent-anchor separation and denoising-trajectory diversion introduced in Section 4.3.2 of the main paper, our adversarial branch includes a cross-attention diversion term. It aims to disturb how the diffusion editor aligns the edit prompt with rendered input, while keeping the editor fixed.

For a sampled view v, we first render the protected scene and the frozen reference scene,

$$I_v^{\text{prot}} = \mathcal{R}_v(\mathcal{G}^{\text{prot}}), \qquad I_v^{\text{ref}} = \mathcal{R}_v(\mathcal{G}^{\text{ref}}), \tag{27}$$

and encode them into the latent space of the diffusion editor,

$$z_v^{\text{prot}} = z(I_v^{\text{prot}}), \qquad z_v^{\text{ref}} = z(I_v^{\text{ref}}). \tag{28}$$

We then construct noisy latents using the same edit prompt ψ, timestep t, and sampled noise ϵ:

$$\tilde{z}_v^{\text{prot}} = \alpha_t z_v^{\text{prot}} + \sigma_t \epsilon, \qquad \tilde{z}_v^{\text{ref}} = \alpha_t z_v^{\text{ref}} + \sigma_t \epsilon. \tag{29}$$

Here, α_t and σ_t are the standard diffusion noise-schedule coefficients at timestep t. Sharing (ψ, t, ϵ) ensures that the resulting diversion signal is driven by the scene difference itself, rather than by stochastic variation inside the editor.

We compute the cross-attention descriptor from the U-Net mid block. Lying at the denoising U-Net's bottleneck, the mid block features have a broader receptive field and encode more semantic, prompt-dependent structure than early high-resolution layers (Jeong et al., 2025). Perturbing the mid-block cross-attention disrupts semantic correspondence between the edit prompt and rendered



(a) Same-view image.

(b) Reference, protected, and difference maps.

Figure 6: **Visualization of cross-attention diversion.** Left: the image from the same viewpoint shown without an attention overlay. Right: visualization of the cross-attention map for *statue*. The three panels overlay the token attention on the frozen reference render, the protected render, and their absolute difference, respectively. The difference panel shows a large attention change around *statue*, indicating that the protected scene changes the editor's prompt-conditioned spatial focus.

image, helping divert the intended editing trajectory while reducing reliance on superficial texture-level perturbations. Let

$$Q^{\text{prot}}_{\text{mid},v} = W^{Q}_{\text{mid}}\big(H_{\text{mid}}(\tilde{z}^{\text{prot}}_{v}, \psi, t)\big), \qquad Q^{\text{ref}}_{\text{mid},v} = W^{Q}_{\text{mid}}\big(H_{\text{mid}}(\tilde{z}^{\text{ref}}_{v}, \psi, t)\big), \tag{30}$$

where $H_{\text{mid}}(\cdot)$ denotes the hidden representation entering the mid-block text-conditioned cross-attention and W^{Q}_{mid} is its query projection. To summarize this response for each rendered view, we average the N_q query tokens, where $Q_{n,:}$ denotes the feature vector of the n-th token:

$$q^{\text{prot}}_{\text{mid},v} = \rho\Big(Q^{\text{prot}}_{\text{mid},v}\Big), \qquad q^{\text{ref}}_{\text{mid},v} = \rho\big(Q^{\text{ref}}_{\text{mid},v}\big), \qquad \rho(Q) = \frac{1}{N_q}\sum_{n=1}^{N_q} Q_{n,:}. \tag{31}$$

We define the cross-attention diversion term for view v as

$$\mathcal{S}^{(v)}_{\text{xattn}} = \frac{1}{d_{\text{mid}}}\left\| q^{\text{prot}}_{\text{mid},v} - q^{\text{ref}}_{\text{mid},v} \right\|_2^2, \tag{32}$$

where d_{mid} is the query dimension of the mid-block descriptor. Since $\mathcal{S}^{(v)}_{\text{xattn}}$ is subtracted in Equation (8), maximizing this diversion term encourages the protected render to induce query representations that differ from those of the frozen reference scene under the same prompt and diffusion condition. This perturbs the prompt-to-image alignment used by the editor and helps divert the editing trajectory.

Importantly, both the diffusion editor and the reference scene remain frozen throughout optimization. Thus, the gradient of $\mathcal{S}^{(v)}_{\text{xattn}}$ flows only through the rendered protected image and ultimately updates the selected Gaussian parameters of $\mathcal{G}^{\text{prot}}$.

Figure 6 provides a visualization of the effect of cross-attention diversion. Figure 6a shows the image from the same viewpoint without an attention overlay. Figure 6b shows the cross-attention map for the prompt *statue*. The third panel in Figure 6b visualizes the difference between the two maps, where red regions indicate areas in which the 3D editing model exhibits larger changes in attention to the given prompt compared with the original image. This reveals a strong change around *statue*, suggesting that the prompt-to-image alignment pathway used by the diffusion editor has been meaningfully altered.

J Additional ablation on masking strategy

In this section, we provide additional ablation results on the Gaussian masking strategy introduced in Section 4.3.1 of the main paper. Instruction-driven 3D editing can target not only salient foreground objects, such as changing the identity, color, or appearance of a main subject, but also background regions, such as altering the scene style, atmosphere, or surrounding environment. In the main paper, we describe an update-saliency-motivated Gaussian selection strategy that identifies a mask-selected Gaussian subset and injects stronger adversarial signals into it. However, the mask is implemented in a soft manner rather than as a strict hard mask, allowing Gaussians that fall below the hard selection threshold but still have nonzero mask overlap to receive weaker adversarial updates.



Figure 7: **Ablation on hard and soft Gaussian masking.** Hard masking confines adversarial updates to the selected region, whereas soft mask concentrates updates on selected Gaussians while retaining adversarial influence on surrounding Gaussians. This design is more suitable for prompts that affect not only foreground objects but also background regions, scene style, or atmosphere.

Figure 7 compares defense performance of hard and soft masking under edits affecting background regions or surrounding scene context. The comparison shows that, in this setting, soft masking is more effective than hard masking because it avoids strictly confining adversarial perturbations to the selected region. Together with the selection ablation in Table 2 of the main paper, this suggests that selecting a Gaussian subset for adversarial noise injection helps balance edit deterrence, watermark recovery, and rendering fidelity, and is better implemented in a soft rather than hard manner.

K Additional ablation on adversarial parameter groups

While the update-saliency-motivated Gaussian selection strategy determines *which Gaussians* should receive adversarial signals, we examine *which parameters* of the selected Gaussians should be perturbed to achieve effective edit deterrence. Following prior 3DGS watermark fine-tuning practice, we keep the mean-position μ_i fixed, since directly changing Gaussian centers can affect scene geometry and rendering quality (Jang et al., 2025). We therefore focus this ablation on the remaining geometry parameters, scale s_i and rotation r_i, and the appearance parameters, opacity α_i and color $\mathbf{c}_i$. Table 9 reports the parameter-group ablation and shows that effective edit deterrence is achieved when adversarial noise is injected into all of these non-position parameter groups.

L Watermark recovery after 3D editing

We further evaluate the watermark recovery performance of existing 3DGS-based watermarking methods after instruction-driven 3D editing (Chen et al., 2024a). For each watermarking baseline, we first protect a pretrained 3DGS scene using the corresponding method, perform editing with DGE (Chen et al., 2024a), and then render the edited scene from the same novel-view evaluation set used in the main experiments. We decode the watermark from these rendered images and report the resulting bit accuracy in Table 10.

As shown in Table 10, existing 3DGS-based watermarking methods exhibit non-negligible drops in bit accuracy after editing. Specifically, 3DGSW, GaussianMarker, and GuardSplat show decreases of 9.68 pp, 23.49 pp, and 44.34 pp, respectively, indicating that the DGE edit-update process can substantially degrade watermark recoverability.

However, the point of this experiment is not merely to measure how well a watermark can be recovered after editing. Even if a watermarking method were to maintain nearly unchanged bit accuracy after 3D editing, watermarking would still remain a passive form of protection that only provides post-hoc ownership evidence. Therefore, practical 3DGS copyright protection requires more than

Table 9: **Ablation on adversarial noise injection across Gaussian parameter groups.** We ablate geometry and appearance parameter groups used for noise injection. Results suggest that injecting adversarial noise into scale, rotation, opacity, and color achieves the strongest edit deterrence while maintaining competitive watermark detectability and rendering quality.

Geometry		Appearance		Bit Acc. (%) ↑	CLIP ↓	PSNR ↑
s_i	r_i	α_i	$\mathbf{c}_i$			
✓	✓	-	-	97.51	0.9658	31.60
-	-	✓	-	97.39	0.9734	<u>31.81</u>
-	-	-	✓	<u>97.54</u>	0.9631	**32.03**
-	-	✓	✓	97.21	0.9486	31.18
✓	✓	-	✓	97.04	<u>0.9322</u>	30.86
✓	✓	✓	-	**97.63**	0.9650	31.03
✓	✓	✓	✓	97.23	**0.9093**	30.36

Table 10: **Bit accuracy after 3D editing.** We report watermark bit accuracy before and after applying DGE (Chen et al., 2024a), an instruction-driven 3D editing method. The Drop (pp) is computed as the decrease from the pre-edit bit accuracy to the post-edit bit accuracy, measured in percentage points.

Drop = Before edit − After edit

Method	Bit Acc. (%) ↑		Drop (pp) ↓
	Before edit	After edit	
3DGSW (Jang et al., 2025)	**99.00**	**89.32**	**9.68**
GaussianMarker (Huang et al., 2024c)	98.51	<u>75.02</u>	<u>23.49</u>
GuardSplat (Chen et al., 2025)	<u>98.92</u>	54.58	44.34

ownership tracing alone. This motivates a unified protection framework that jointly optimizes recoverable ownership tracing and active edit deterrence within a single protected 3DGS representation.

M ADDITIONAL QUALITATIVE RESULTS

Figure 8 shows additional qualitative results beyond those shown in the main paper.

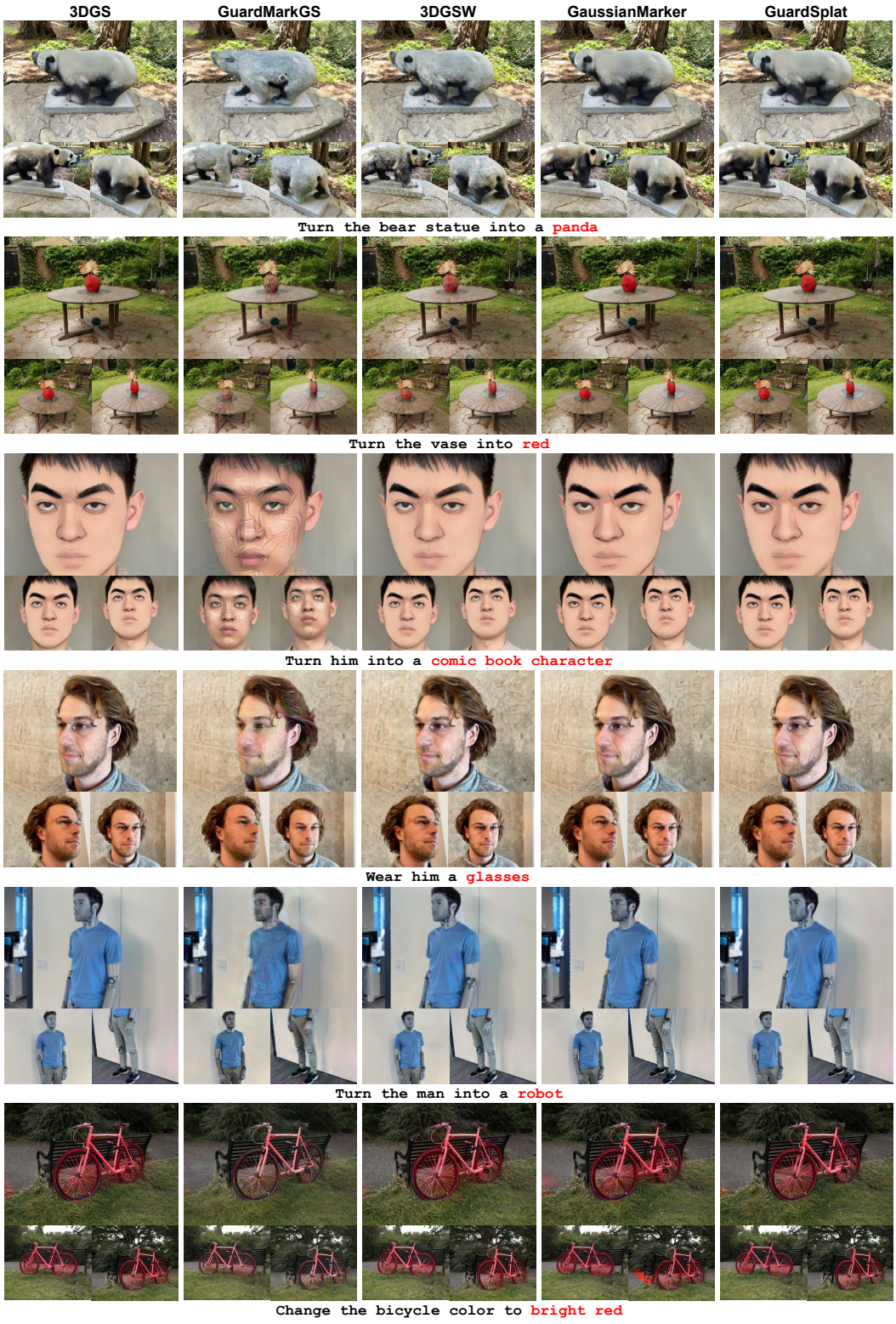


Figure 8: **Additional qualitative results.** We show the editing results for each scene using DGE (Chen et al., 2024a).