RESOLVING SURROGATE TARGET AMBIGUITY IN BLACK-BOX 3D GAUSSIAN SPLATTING WATERMARK ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermark robustness against adversarial attacks is important for protecting 3D Gaussian Splatting (3DGS) assets. Existing black-box attacks primarily rely on indirect derivative-free optimization without exploiting differentiable surrogates, such as public decoders, resulting in high computational cost, substantial modifications to the 3DGS representation, and limited rendering fidelity. Although the differentiable renderer and public decoder provide actionable surrogate gradients, they do not reveal which surrogate target will effectively disrupt the hidden owner decoder. We refer to this problem as Surrogate Target Ambiguity, which can cause unstable attack effectiveness as surrogate–owner alignment varies. To enable stronger evaluation of 3DGS watermark robustness, we propose Alignment-Agnostic Carrier Targeting (AACT), which constructs an effective attack target without relying on the unknown surrogate–owner alignment. AACT combines a fixed half-Hamming displacement, which avoids alignment-dependent endpoint selection, with carrier-aware bit selection based on strength and complementarity to construct an asset-conditioned target. The target is then directly optimized through bounded appearance-space perturbations while preserving Gaussian geometry and primitive structure. Experiments on three representative 3DGS watermarking methods, GaussianMarker, GuardSplat, and 3D-GSW, show that the proposed method achieves strong attack effectiveness while substantially improving rendering fidelity, efficiency, and structural preservation compared with the representative black-box attack GMEA. In particular, it preserves all Gaussian primitives while reducing attack time from thousands of seconds to tens of seconds. Code will be made publicly available.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) represents a scene as an explicit set of Gaussian primitives that can be efficiently rendered, distributed, and directly modified (Kerbl et al., 2023). While these properties make 3DGS attractive for sharing and editing 3D assets, they also expose the underlying representation to unauthorized copying and manipulation, motivating effective watermarking methods (Huang et al., 2024; Jang et al., 2025; Chen et al., 2025). However, robustness to predefined perturbations, such as compression, editing, and fixed transformations, does not necessarily guarantee security against adversarial attacks that intentionally disrupt watermark verification while preserving the protected asset. This motivates the study of watermark attacks to evaluate the security of 3DGS watermarking systems (Huang et al., 2026; Zeng et al., 2026). In particular, black-box attacks operate in a challenging setting where the attacker can access and modify the watermarked 3DGS but has no access to the owner decoder, embedded message, or owner-side verification process. A strong attack should therefore effectively disrupt the watermark while maintaining high rendering fidelity and preserving the underlying Gaussian structure.

Existing black-box 3DGS watermark attacks remain limited in their ability to meet these requirements. A representative example is GMEA (Zeng et al., 2026), which formulates the attack as a multi-objective evolutionary search over Gaussian pruning and color perturbation, using indirect watermark evidence to evaluate candidate modifications. However, as shown in Figure 1(a), this derivative-free formulation requires repeated candidate rendering and evaluation, incurring high computational cost, while Gaussian pruning can substantially alter the original 3DGS representation and degrade rendering fidelity. Importantly, the inaccessibility of the owner decoder only prevents

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.

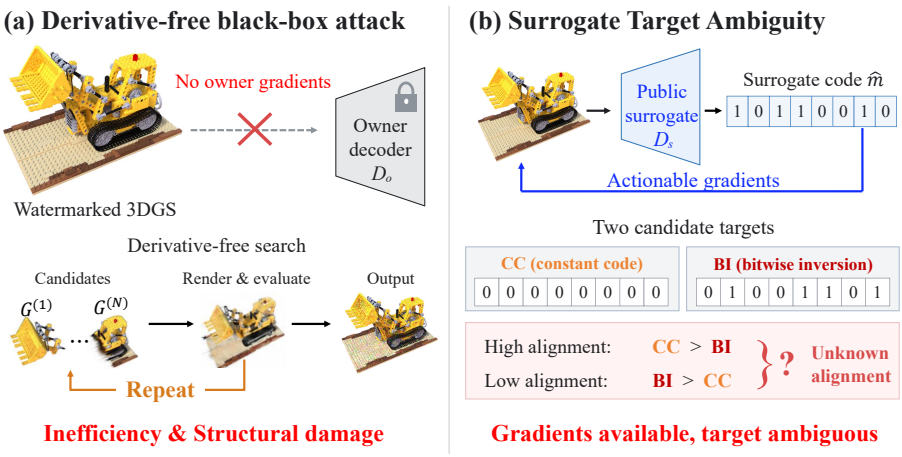


Figure 1: **Motivation and core idea.** (a) GMEA relies on derivative-free search and alters the Gaussian representation through pruning. (b) Public-surrogate gradients do not specify an attack target, and CC/BI preferences differ across the evaluated decoder settings.

gradients through the protected verification path; it does not imply that useful differentiable guidance is unavailable. A differentiable 3DGS renderer together with an independently available public decoder can instead provide surrogate gradients for directly optimizing the 3DGS representation. This raises a natural question: can such surrogate gradients enable more effective, efficient, and structure-preserving black-box 3DGS watermark attacks?

Prior studies on transfer-based attacks have shown that surrogate-derived optimization directions can remain useful even when the inaccessible target model differs from the surrogate model (Wang et al., 2021; Zhang et al., 2022). However, surrogate gradients only indicate how to optimize a chosen target; they do not determine which surrogate target will most effectively disrupt the hidden owner decoder. Although the public decoder provides an observable surrogate pseudo-code, the corresponding attack target remains ambiguous because the latent bitwise correspondence between the surrogate and owner decoders, referred to as surrogate-owner alignment, is unknown. As illustrated in Figure 1(b), this ambiguity is exposed by two simple endpoint targets. Constant-Code (CC) drives the surrogate output toward a fixed codeword, whereas Bitwise-Inversion (BI) flips every surrogate pseudo-code bit. Empirically, CC is substantially stronger under high surrogate-owner alignment, whereas BI becomes more effective under lower alignment. Since this alignment is unavailable to the attacker, choosing the more effective target would require inaccessible information. We refer to this challenge as **Surrogate Target Ambiguity**.

To address this challenge, we propose a differentiable black-box attack built around **Alignment-Agnostic Carrier Targeting (AACT)**. AACT constructs an attack target without estimating or relying on the unknown surrogate–owner alignment. It first imposes a fixed half-Hamming displacement from the surrogate pseudo-code, providing an alignment-agnostic target geometry that avoids choosing between alignment-dependent endpoints. Since many targets satisfy this geometry, AACT further uses bit-specific gradients over Gaussian appearance parameters to identify bits that are both easy to manipulate and supported by complementary Gaussian carriers. A balanced D-optimal criterion then selects these bits to form an asset-conditioned target. The resulting target is directly optimized through bounded appearance-space perturbations while keeping the Gaussian geometry and primitive set unchanged. Finally, **Self-Normalized Saturation Stopping (SNSS)** adapts the stopping point to each asset's optimization trajectory, avoiding unnecessary optimization after the target response saturates.

Our main contributions are summarized as follows:

- We identify **Surrogate Target Ambiguity** as a key challenge in differentiable black-box 3DGS watermark attacks: although surrogate gradients provide actionable guidance, they do not specify which surrogate target will most effectively disrupt the hidden owner decoder when surrogate-owner alignment is unknown.

- We propose **Alignment-Agnostic Carrier Targeting (AACT)**, which constructs asset-conditioned attack targets without estimating the latent surrogate-owner alignment. AACT combines a fixed half-Hamming target geometry with balanced carrier-aware bit selection that favors accessible and complementary bits, followed by bounded appearance-space optimization that preserves Gaussian geometry and primitive structure.
- Experiments on GaussianMarker, GuardSplat, and 3D-GSW validate Surrogate Target Ambiguity across different surrogate-owner alignment settings and demonstrate the effectiveness of AACT without alignment-dependent target selection. Compared with GMEA, our attack preserves all Gaussian primitives, substantially improves rendering fidelity, and reduces attack time from thousands of seconds to tens of seconds.

2 Related Work

3DGS Watermarking. Recent 3DGS watermarking methods protect scene ownership through diverse designs, including dedicated carriers, Gaussian restructuring, spherical-harmonic perturbation, and secured appearance representations (Huang et al., 2024; Jang et al., 2025; Chen et al., 2025; Zhang et al., 2024). Subsequent work improves generalizable embedding, robustness to compression and editing, adaptive watermark placement, and native 3D protection (Li et al., 2025; Huang et al., 2025; In et al., 2026; Zhao et al., 2025; Guo et al., 2026; Qin et al., 2026; Li et al., 2026; Cai et al., 2026; Wang et al., 2026). For black-box attacks, an important source of heterogeneity lies in the owner-side decoder. Existing systems range from designs that reuse generic pretrained image-watermark decoders to those employing method-specific decoders coupled to their watermark representations and training objectives. GaussianMarker (Huang et al., 2024) and GuardSplat (Chen et al., 2025) provide representative examples of these two designs, respectively. Consequently, the bitwise correspondence between an independently available public surrogate and the inaccessible owner decoder cannot be assumed known a priori.

Watermark Attacks. Image-watermark removal spans adversarial evasion, regeneration, averaging, universal and semantic suppression, and no-box or black-box attacks under diverse access assumptions (Jiang et al., 2023; Zhao et al., 2024; Yang et al., 2024; Kassis & Hengartner, 2025; Qiu et al., 2026; Hu et al., 2025; Müller et al., 2025; An et al., 2026a;b). Related adversarial-transfer studies further show that surrogate-derived representations can provide transferable guidance across models (Huang et al., 2019; Lu et al., 2020; Wang et al., 2021; Zhang et al., 2022). These methods are primarily formulated in 2D image space or reconstruction pipelines and do not address how an attack target should be constructed when the surrogate-owner alignment is unknown during direct optimization of a multi-view explicit 3D representation.

Dedicated attacks on 3DGS watermarks are more recent. GSPure (Huang et al., 2026) localizes and removes watermark-related Gaussian primitives, while GMEA (Zeng et al., 2026) performs black-box removal without target-decoder access using an indirect feature-dispersion objective and evolutionary optimization over Gaussian pruning and color perturbation. These methods establish the feasibility of dedicated 3DGS attacks, but do not exploit an independent public surrogate for differentiable optimization or address attack-target construction under unknown surrogate-owner alignment. Our work studies this unexplored regime and develops an alignment-agnostic solution.

3 Method

As illustrated in Figure 2, our black-box 3DGS watermark attack consists of three stages. **Surrogate Setup** defines the threat model and derives a surrogate decoding objective from a public decoder. **Target Construction** builds an alignment-agnostic, asset-conditioned attack target without assuming knowledge of surrogate–owner alignment. **Attack Optimization** updates only Gaussian appearance parameters under bounded perturbations while preserving geometry and primitives, with SNSS adaptively determining when to stop.

3.1 Surrogate Setup

Let $\mathcal{G}^w$ denote a watermarked 3D Gaussian representation and $\mathcal{R}(\mathcal{G}, \phi)$ its differentiable rendering at camera pose ϕ. We consider a black-box attacker with access to $\mathcal{G}^w$, a set of rendering poses Φ,

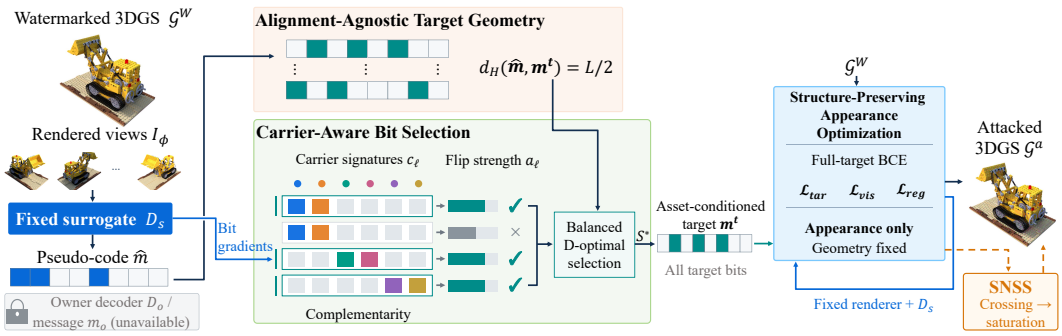


Figure 2: **Overview of the proposed AACT framework.**

and an independently available fixed public decoder $\mathcal{D}_s$. The owner-side decoder $\mathcal{D}_o$, the embedded message $\mathbf{m}_o$, and any owner-side verification queries are inaccessible. The goal is to obtain an attacked representation $\mathcal{G}^a$ that disrupts owner-side watermark verification while preserving the rendered appearance of $\mathcal{G}^w$. For each pose $\phi \in \Phi$, the public decoder produces the pre-sigmoid surrogate logits $\mathbf{z}^s_\phi = \mathcal{D}_s(\mathcal{R}(\mathcal{G}^w, \phi)) \in \mathbb{R}^L$. We aggregate them across views as

$$\bar{\mathbf{z}}^s = \frac{1}{|\Phi|} \sum_{\phi \in \Phi} \mathbf{z}^s_\phi, \qquad \hat{\mathbf{m}} = \mathbb{1}[\bar{\mathbf{z}}^s > 0], \tag{1}$$

where $\mathbb{1}[\cdot]$ denotes the element-wise indicator function and $\hat{\mathbf{m}} \in \{0,1\}^L$ is the observable *surrogate pseudo-code*. We distinguish $\hat{\mathbf{m}}$ from the attack target $\mathbf{m}^t$. The former is the current prediction of the public decoder, whereas the latter specifies the output toward which the attack is optimized. Although the differentiable renderer and D_s provide actionable gradients, the attacker does not know the latent bitwise correspondence between D_s and D_o, referred to as *surrogate–owner alignment*. Consequently, $\hat{\mathbf{m}}$ alone does not determine which target $\mathbf{m}^t$ will most effectively disrupt the hidden owner decoder. We refer to this ambiguity as surrogate target ambiguity.

3.2 Target Construction

Given the surrogate pseudo-code $\hat{\mathbf{m}}$, we construct an attack target $\mathbf{m}^t$ without estimating the unknown surrogate-owner alignment. AACT consists of two steps. First, we define an alignment-agnostic target geometry that fixes how far the attack target should move from the surrogate pseudo-code. Second, among the many targets satisfying this geometry, we select the bits that are both easy to manipulate and supported by complementary Gaussian carriers. The resulting asset-conditioned target remains fixed during the subsequent attack optimization.

Alignment-Agnostic Target Geometry. Let $S \subseteq \{1, \ldots, L\}$ denote the set of surrogate bits to be flipped. Because the surrogate-owner alignment is unknown, directly choosing an endpoint such as Constant-Code or Bitwise-Inversion can lead to alignment-dependent attack effectiveness. We therefore avoid endpoint selection and instead constrain the attack target to differ from the surrogate pseudo-code in exactly half of its bits:

$$m^t_\ell = \begin{cases} 1 - \hat{m}_\ell, & \ell \in S, \\ \hat{m}_\ell, & \ell \notin S, \end{cases} \qquad |S| = \frac{L}{2}, \tag{2}$$

which gives $d_H(\hat{\mathbf{m}}, \mathbf{m}^t) = L/2$. Using the signed binary encoding $\mathbf{s}(\mathbf{m}) = 2\mathbf{m} - \mathbf{1}$, we obtain $\mathbf{s}(\hat{\mathbf{m}})^\top \mathbf{s}(\mathbf{m}^t) = L - 2d_H(\hat{\mathbf{m}}, \mathbf{m}^t) = 0$. Thus, the signed surrogate pseudo-code and attack target are orthogonal in code space. We use this half-Hamming condition as a fixed, alignment-agnostic displacement that does not depend on the unknown surrogate-owner relationship. We do not assume that half-Hamming displacement is universally optimal; rather, it provides a single relation-independent geometry for the considered black-box setting. Although this geometry removes alignment-dependent endpoint selection, it still leaves $\binom{L}{L/2}$ admissible targets. The remaining question is therefore which bits should be flipped for the current 3DGS asset.

Carrier-Aware Bit Selection. Not all admissible bit subsets are equally realizable. Different surrogate bits vary in their sensitivity to appearance perturbations, and multiple bits may rely on highly

overlapping Gaussian carriers. We therefore characterize each bit using two complementary properties: *flip strength*, which measures how easily the bit can be changed through appearance perturbations, and a *carrier signature*, which describes where its response is concentrated over the Gaussian set.

Let the watermarked representation contain N Gaussians, with $\Theta^{\text{app}} = \{\Theta_i^{\text{app}}\}_{i=1}^N$ denoting all trainable appearance parameters and Θ_i^{app} the parameters of Gaussian i. Let $\Phi_c \subseteq \Phi$ be a small set of probe views. For bit ℓ and view ϕ, we define the flip-oriented gradient and its Gaussian-wise response as

$$\mathbf{g}_{\ell,i}^{\phi} = \nabla_{\Theta_i^{\text{app}}} \left[(1 - 2\hat{m}_\ell) z_{\phi,\ell}^s \right], \qquad e_{\ell,i}^{\phi} = \left\| \mathbf{g}_{\ell,i}^{\phi} \right\|_2. \tag{3}$$

The factor $1 - 2\hat{m}_\ell$ orients the gradient toward flipping the current surrogate decision. For each probe view, let $\mathcal{T}_\ell^\phi$ contain the strongest Gaussian responses for bit ℓ. We aggregate their normalized responses into a *carrier signature* $\mathbf{c}_\ell \in \mathbb{R}^N$ and define the overall flip strength a_ℓ as

$$c_{\ell,i} = \frac{1}{|\Phi_c|} \sum_{\phi \in \Phi_c} \frac{\mathbb{1}[i \in \mathcal{T}_\ell^\phi]\, e_{\ell,i}^\phi}{\sum_{j \in \mathcal{T}_\ell^\phi} e_{\ell,j}^\phi + \epsilon}, \qquad a_\ell = \frac{1}{|\Phi_c|} \sum_{\phi \in \Phi_c} \left\| \nabla_{\Theta^{\text{app}}} \left[(1 - 2\hat{m}_\ell) z_{\phi,\ell}^s \right] \right\|_2, \tag{4}$$

where $\epsilon > 0$ is a numerical stabilizer. The flip strength a_ℓ measures the overall accessibility of bit ℓ, while the carrier signature $\mathbf{c}_\ell$ captures the Gaussian support responsible for its response. Selecting bits solely according to a_ℓ can be redundant because several strong bits may rely on nearly identical Gaussian carriers. We therefore seek bits that are both accessible and complementary.

To jointly account for these properties, we convert the flip strengths into bounded weights $w_\ell = [\text{clip}(a_\ell/\bar{a}, a_{\min}, a_{\max})]^\beta$, where $\bar{a}$ is the median positive strength and $a_{\min}$, $a_{\max}$, and β are fixed hyperparameters specified in Table 4. We then construct the weighted carrier kernel

$$K_{\ell k} = w_\ell w_k \frac{\mathbf{c}_\ell^\top \mathbf{c}_k}{\|\mathbf{c}_\ell\|_2 \|\mathbf{c}_k\|_2 + \epsilon}. \tag{5}$$

The strength weights favor readily manipulable bits, while the carrier-signature similarity measures the overlap between their Gaussian support. We then greedily select a balanced subset using a D-optimal criterion (Singh & Xie, 2020):

$$S^\star \approx \arg \max_{\substack{|S|=L/2 \\ S \in \mathcal{B}(\hat{\mathbf{m}})}} \log\det(\mathbf{I} + \gamma \mathbf{K}_S), \tag{6}$$

where $\mathcal{B}(\hat{\mathbf{m}})$ enforces an approximately balanced number of pseudo-zero and pseudo-one flips, preventing a half-Hamming target from degenerating toward a constant-code extreme. The strength weighting favors readily manipulable bits, while the log-determinant discourages redundant carrier signatures and promotes complementary Gaussian support. Substituting the selected set $S^\star$ into Eq. (2) yields the final asset-conditioned target $\mathbf{m}^t$. Appendix B provides an example.

3.3 Attack Optimization

Given the asset-conditioned target $\mathbf{m}^t$ constructed by AACT, we directly optimize the watermarked 3DGS toward this target while preserving its rendered appearance and underlying structure. We modify only the Gaussian appearance parameters, keeping the Gaussian geometry and primitive set fixed. To avoid unnecessary perturbations after the target has been sufficiently reached, we further introduce self-normalized saturation stopping (SNSS), which adapts the stopping point to each asset's optimization trajectory.

Structure-Preserving Appearance Optimization. At each iteration, let $\mathcal{V} \subseteq \Phi$ denote the sampled rendering views. We optimize all surrogate-code coordinates toward the fixed target $\mathbf{m}^t$ using the per-view binary cross-entropy

$$\mathcal{L}_{\text{tar}} = \frac{1}{|\mathcal{V}|} \sum_{\phi \in \mathcal{V}} \text{BCE}\left(\mathbf{z}_\phi^s, \mathbf{m}^t\right). \tag{7}$$

Here, $z_\phi^s = D_s(R(\mathcal{G}, \phi))$ denotes the surrogate logits of the current representation $\mathcal{G}$. Unlike pruning-based attacks, we optimize only the Gaussian appearance parameters Θ^{app} while keeping the Gaussian geometry and primitive set unchanged. The overall attack objective is

$$\mathcal{L}_{\text{attack}} = \lambda_{\text{tar}} \mathcal{L}_{\text{tar}} + \lambda_{\text{vis}} \mathcal{L}_{\text{vis}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \tag{8}$$

where $\mathcal{L}_{\text{tar}}$ drives the public decoder toward the constructed target, $\mathcal{L}_{\text{vis}}$ preserves multi-view rendering consistency between the attacked and original watermarked representations, and $\mathcal{L}_{\text{reg}}$ penalizes excessive deviations of the appearance parameters from their original values.

To further constrain the attack, each trainable appearance tensor Θ_q^{app} is projected after every update into a bounded neighborhood of its original value:

$$\Theta_q^{\text{app}} \leftarrow \text{clip}\left(\Theta_q^{\text{app}}, \Theta_q^{\text{app},0} - r_q\sigma_q, \Theta_q^{\text{app},0} + r_q\sigma_q\right), \tag{9}$$

where $\Theta_q^{\text{app},0}$ denotes the original parameter tensor, σ_q its standard deviation, and r_q a fixed relative radius. This bounded appearance-space optimization avoids Gaussian pruning, insertion, and geometric modification, thereby preserving both the primitive set and Gaussian geometry. Losses and optimization details are provided in Appendix A.2.

Self-Normalized Saturation Stopping (SNSS). Different 3DGS assets reach the surrogate target at different rates. A fixed iteration budget can therefore terminate too early before the target is sufficiently reached, or continue optimization after the attack has saturated, introducing unnecessary perturbations without consistent additional benefit. SNSS determines the stopping point from the optimization trajectory of each asset.

Let $\Phi_p \subseteq \Phi$ be a fixed set of public probe views and $\mathcal{A} = S^\star$ the bits selected by AACT. At probe step j, we compute the full-target probe loss and the minimum signed margin over the selected bits:

$$\mathcal{P}_j = \frac{1}{|\Phi_p|}\sum_{\phi\in\Phi_p}\text{BCE}\left(\mathbf{z}_{\phi,j}^s, \mathbf{m}^t\right), \quad \mu_j = \min_{\ell\in\mathcal{A}}\left[(2m_\ell^t - 1)\bar{z}_{j,\ell}^s\right], \quad \bar{\mathbf{z}}_j^s = \frac{1}{|\Phi_p|}\sum_{\phi\in\Phi_p}\mathbf{z}_{\phi,j}^s. \tag{10}$$

The target boundary is considered crossed when $\mu_j > 0$, indicating that all AACT-selected bits have reached the target side under the aggregated probe views. After this crossing, SNSS monitors the relative decrease in probe loss:

$$r_j = \frac{\mathcal{P}_{j-1} - \mathcal{P}_j}{|\mathcal{P}_{j-1}| + \epsilon}. \tag{11}$$

Rather than comparing r_j with a globally calibrated threshold, SNSS uses the median of the first few post-crossing rates as an asset-specific reference r_{ref}. Optimization terminates when subsequent improvement remains below a fixed fraction of r_{ref} for consecutive probe intervals while μ_j remains positive. This self-normalized criterion adapts to varying convergence rates across assets and avoids both premature stopping and unnecessary over-optimization without requiring owner-side verification or cross-asset calibration.

4 Experiment

4.1 Experimental Setup

Watermarking Systems and Datasets. We evaluate our attack on three representative 3DGS watermarking systems: GaussianMarker (Huang et al., 2024), GuardSplat (Chen et al., 2025), and 3D-GSW (Jang et al., 2025). GaussianMarker and GuardSplat serve as the primary evaluation systems because they exhibit substantially different surrogate-owner alignment, while 3D-GSW provides an additional high-alignment setting. Experiments are conducted on 15 scenes from the Blender (Mildenhall et al., 2021) and LLFF (Mildenhall et al., 2019) datasets. All compared attacks use the same scene set and evaluation views.

We characterize surrogate-owner alignment by the average bitwise agreement between the aggregated codes decoded by the public surrogate $\mathcal{D}_s$ and the owner decoder $\mathcal{D}_o$ on the same watermarked assets. Higher agreement indicates stronger alignment. This statistic is computed only for post-hoc analysis and is never available to the attacker during optimization. GaussianMarker and 3D-GSW exhibit complete bitwise agreement with the public surrogate, with alignment values of 100.0%, whereas GuardSplat has substantially lower alignment at 54.02%. These settings allow us to evaluate attack-target construction under markedly different surrogate-owner relationships.

Baseline Attacks. We compare against GMEA (Zeng et al., 2026), a dedicated black-box 3DGS watermark attack applicable to the render-decoded bitstream setting considered here. We reproduce GMEA using the authors' released implementation and retain its original attack-specific settings.

To isolate the effect of surrogate target construction, we additionally consider two endpoint baselines: *Constant-Code* (CC), which drives the surrogate output toward a fixed codeword, and *Bitwise-Inversion* (BI), which uses the bitwise complement of the surrogate pseudo-code as the target.

Evaluation Metrics. We evaluate watermark disruption using Bit Accuracy Rate (BAR) and Information Destruction Score (IDS) (Zeng et al., 2026). For a binary watermark, BAR approaching 50% indicates random-level recovery. To make this deviation explicit, we additionally report $\Delta\mathrm{BAR} = |\mathrm{BAR} - 50|$, where lower values indicate stronger disruption. Higher IDS likewise indicates greater information destruction. Rendering fidelity is evaluated using PSNR and SSIM, with higher values indicating better preservation. We further report attack time and the retained-Gaussian ratio (*Keep*) to measure computational efficiency and structural preservation, respectively.

Implementation Details. We use a fixed pretrained 48-bit HiDDeN decoder (Zhu et al., 2018) as the public surrogate for all surrogate-guided attacks. Our method modifies only Gaussian appearance parameters and uses Adam with a learning rate of 4×10^{-3} and a maximum budget of 80 optimization steps. Each update uses eight randomly sampled rendering views through gradient accumulation, while four additional probe views are used to construct the carrier signatures for AACT. CC, BI, and AACT share the same appearance-space optimization and SNSS protocol and differ only in target construction. All experiments are repeated using three fixed random seeds, and results are averaged across seeds. No owner decoder, embedded message, or owner-side verification query is accessed during attack optimization. Additional loss weights, probe settings, projection radii, and stopping hyperparameters are provided in Appendix A.

Table 1: **Comparison with black-box 3DGS watermark attack baselines.**

Watermark	Attack	ΔBAR ↓	IDS ↑	PSNR ↑	SSIM ↑	Time (s) ↓	Keep ↑
GaussianMarker	No attack	49.984	0.031	27.072	0.8993	–	1.000
	GMEA	18.709	62.599	23.332	0.7474	5728.9	0.483
	AACT	**0.000**	**99.826**	**25.481**	**0.8337**	**45.9**	**1.000**
GuardSplat	No attack	46.256	7.454	32.397	0.9660	–	1.000
	GMEA	**6.979**	**82.858**	22.305	0.7366	2181.2	0.482
	AACT	9.361	79.692	**28.604**	**0.9016**	**35.1**	**1.000**
3D-GSW	No attack	48.719	2.514	28.156	0.9223	–	1.000
	GMEA	17.169	65.636	22.204	0.7296	2531.8	0.485
	AACT	**3.248**	**93.415**	**24.716**	**0.8330**	**41.7**	**1.000**

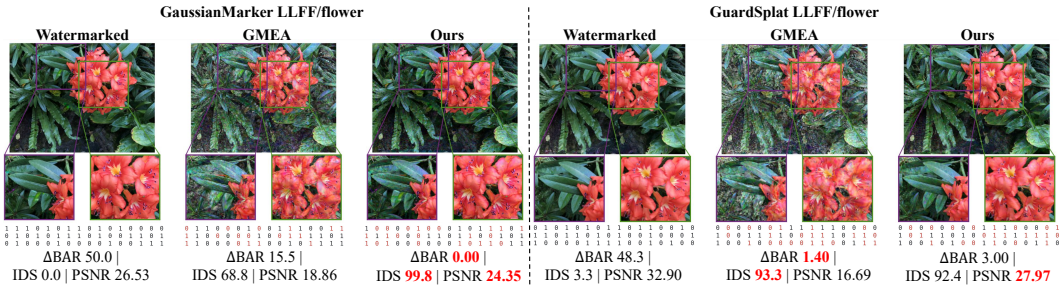


Figure 3: **Qualitative comparison on the LLFF Flower scene.** Watermarked renders, GMEA, and our attack are shown for GaussianMarker and GuardSplat, with enlarged regions for local comparison. CC and BI are omitted because they share the same appearance-space optimization and exhibit comparable rendering fidelity.

4.2 Main Results

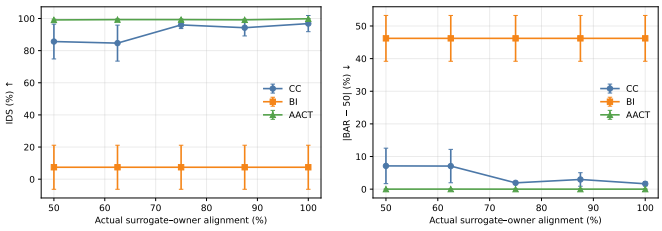
Comparison with Black-Box Attack Baselines. As shown in Table 1, AACT achieves substantially stronger watermark disruption than GMEA on GaussianMarker and 3D-GSW, reducing $\Delta\mathrm{BAR}$ from 18.709 and 17.169 to 0.000 and 3.248, while increasing IDS from 62.599 and 65.636 to 99.826 and 93.415, respectively. On GuardSplat, GMEA achieves slightly stronger disruption, with $\Delta\mathrm{BAR} = 6.979$ and $\mathrm{IDS} = 82.858$, compared with 9.361 and 79.692 for AACT. However, AACT provides markedly better preservation of the underlying 3DGS asset, improving PSNR from 22.305

Table 2: **Comparison across different surrogate-owner alignment settings.** Each system reports ΔBAR/IDS, with cross-setting metrics averaged over all three systems.

	High Alignment		Low Alignment	Cross-Setting Average			
Method	GaussianMarker	3D-GSW	GuardSplat	ΔBAR ↓	IDS ↑	PSNR ↑	SSIM ↑
CC	1.567 / 96.821	6.311 / 78.247	12.180 / 71.348	6.686	82.139	26.262	**0.8626**
BI	46.210 / 7.418	28.507 / 42.465	**7.226 / 81.441**	27.314	43.775	26.007	0.8545
AACT	**0.000 / 99.826**	**3.248 / 93.415**	9.361 / 79.692	**4.203**	**90.978**	**26.267**	0.8561

to 28.604 dB and SSIM from 0.7366 to 0.9016, while retaining all Gaussian primitives compared with only 48.2% for GMEA. The qualitative results in Figure 3 are consistent with these fidelity differences: in the enlarged regions, GMEA introduces more noticeable local appearance degradation, whereas AACT better preserves texture and color structure on both GaussianMarker and GuardSplat. AACT is also substantially more efficient, reducing attack time from 2181.2–5728.9 seconds for GMEA to 35.1–45.9 seconds. Overall, AACT achieves strong watermark disruption while providing substantially better rendering fidelity, structural preservation, and computational efficiency. Appendix E provides qualitative comparisons across all evaluated Blender and LLFF scenes and discussions on the challenging Room scene.

Comparison of Different Target Construction Strategies. Table 2 compares different target constructions across surrogate–owner alignment settings. On the two high-alignment systems, GaussianMarker and 3D-GSW, CC substantially outperforms BI, achieving ΔBAR/IDS of 1.567/96.821 and 6.311/78.247, respectively, compared with 46.210/7.418 and 28.507/42.465 for BI. This trend reverses on the lower-alignment GuardSplat setting, where BI improves ΔBAR from 12.180 to 7.226 and IDS from 71.348 to 81.441. These results show that the effectiveness of a fixed endpoint target depends strongly on the surrogate-owner relationship, which is unavailable to the attacker. In contrast, AACT avoids alignment-dependent endpoint selection and achieves the strongest cross-setting attack effectiveness, with an average ΔBAR of 4.203 and IDS of 90.978, while maintaining comparable rendering fidelity. This behavior empirically supports Surrogate Target Ambiguity and motivates alignment-agnostic target construction.



(a) IDS. (b) ΔBAR.

Figure 4: **Controlled surrogate–owner alignment.** Under fixed assets and attack protocol, (a) IDS and (b) ΔBAR show that AACT remains effective from 100% to 50% alignment. Error bars denote standard deviation across correspondence masks.

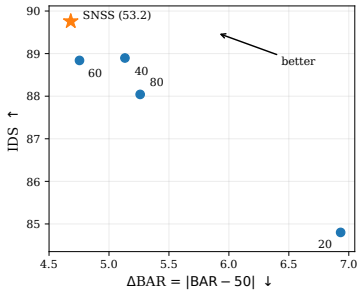


Figure 5: **Ablation of SNSS.** Fixed budgets use 20–80 steps, whereas SNSS stops at 53.2 steps on average.

Controlled Surrogate-Owner Alignment. To isolate the effect of surrogate-owner correspondence from other differences between watermarking systems, we conduct a controlled study on GaussianMarker, whose original surrogate-owner alignment is 100%. We keep the watermarked assets, owner decoder, and attack protocol fixed, and construct controlled public surrogates by applying a fixed bit-polarity mapping to the surrogate logits, $\tilde{z}^s_\ell = q_\ell z^s_\ell$ with $q_\ell \in \{-1, +1\}$. By varying the flipped coordinates, we obtain alignment levels ranging from 100% to 50%. As shown in Figure 4, AACT remains consistently effective across this range: IDS varies only from 99.826 to 99.111, while ΔBAR remains near zero throughout. In contrast, CC degrades as alignment decreases and exhibits substantially larger variation across polarity mappings. BI is nearly invariant to the remapping but consistently yields low IDS and large ΔBAR. These results further show that AACT remains effective even with substantial changes in surrogate-owner bitwise correspondence, without estimating the latent alignment.

Table 3: **Ablation of AACT.** (a) Target displacement ratio ρ. (b) Bit selection under half-Hamming geometry: Diversity uses complementarity, Strength uses carrier strength, and Full combines both.

(a) Target geometry

ρ	ΔBAR ↓	IDS ↑	Worst IDS ↑	PSNR ↑
0.25	19.202	60.818	50.087	**27.828**
0.50	**4.681**	**89.759**	**72.996**	27.043
0.75	7.874	64.109	49.739	27.077
1.00	26.718	44.430	0.000	26.765

(b) Bit selection

Selection	ΔBAR ↓	IDS ↑	Succ. ↑	Iter. ↓	PSNR ↑
Random	6.203	87.275	91.7	29.53	27.307
Diversity	5.584	<u>87.946</u>	81.2	34.06	**27.425**
Strength	5.131	87.480	**100.0**	17.23	26.976
Full	**4.681**	**89.759**	**100.0**	**15.54**	27.043

4.3 Ablation Studies

We conduct ablation studies on GaussianMarker and GuardSplat, representing high- and low-alignment settings, respectively. Unless otherwise specified, results are averaged over the two systems. In each study, only the component under investigation is varied, while all other settings are kept fixed.

Alignment-Agnostic Target Geometry. We first vary the target displacement ratio ρ, defined as the fraction of surrogate pseudo-code bits to be flipped, while keeping all other components fixed. As shown in Table 3(a), the half-Hamming setting ($\rho = 0.5$) achieves the lowest average $\Delta\mathrm{BAR}$ of 4.681, the highest average IDS of 89.759, and the highest worst-case IDS of 72.996 over all scenes of GaussianMarker and GuardSplat. A smaller displacement ($\rho = 0.25$) provides insufficient disruption, whereas larger displacements reduce cross-setting robustness: the worst-case IDS drops to 49.739 at $\rho = 0.75$ and to 0.000 under full inversion ($\rho = 1.0$). These results support half-Hamming displacement as a stable alignment-agnostic target geometry when surrogate-owner alignment is unknown.

Carrier-Aware Bit Selection. We next fix the half-Hamming geometry and balanced flip constraint and vary only the bit-selection rule. Random ignores carrier information, Diversity uses only carrier complementarity, Strength uses only carrier strength, and Full combines both through the D-optimal objective in Sec. 3.2. As shown in Table 3(b), Random achieves a target-crossing success rate of 91.7% with 29.53 iterations, whereas Diversity reduces the success rate to 81.2%, indicating that complementarity alone does not ensure bit accessibility. Strength restores 100% success and reduces the crossing cost to 17.23 iterations. Full further achieves the lowest $\Delta\mathrm{BAR}$ of 4.681, the highest IDS of 89.759, and the fewest crossing iterations of 15.54. These results show that carrier strength is critical for selecting accessible bits, while carrier complementarity reduces redundancy and further improves attack effectiveness.

Self-Normalized Saturation Stopping. Figure 5 compares SNSS with otherwise identical attacks using fixed budgets of 20, 40, 60, and 80 optimization steps. A 20-step budget clearly under-optimizes the attack, whereas extending optimization beyond 40–60 steps does not consistently improve owner-side disruption. In particular, the 80-step setting degrades to $\Delta\mathrm{BAR} = 5.260$ and IDS $= 88.042$. In contrast, SNSS stops after 53.2 steps on average and achieves the lowest $\Delta\mathrm{BAR}$ of 4.681 and the highest IDS of 89.759. These results show that adapting the stopping point to each asset's optimization trajectory avoids both premature termination and unnecessary over-optimization.

5 Conclusion

We revisit the security evaluation of black-box 3DGS watermarking and show that owner-side decoder inaccessibility does not eliminate differentiable attack guidance: an independent public surrogate can still provide actionable gradients. The resulting challenge is Surrogate Target Ambiguity, where unknown surrogate-owner alignment leaves attack-target construction underdetermined. We address this challenge with Alignment-Agnostic Carrier Targeting (AACT), which combines a fixed half-Hamming geometry with carrier-aware bit selection to construct asset-conditioned targets without estimating the latent alignment, followed by bounded appearance-space optimization and SNSS. Experiments on GaussianMarker, GuardSplat, and 3D-GSW validate this ambiguity and show that AACT effectively disrupts watermarks across different alignment settings while substantially improving rendering fidelity, structural preservation, and efficiency compared to the prior derivative-free attack.

AI USE STATEMENT

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts and ideas were developed by the authors. Contributions of the LLM focused solely on improving the linguistic quality of the article, without involvement in the scientific content or data analysis.

We take full responsibility for the final content of this work, including the methodology, scientific claims, experimental results, code, and figures. This responsibility includes reviewing AI-assisted content, validating code, and checking reported results against the original experimental records.

ETHICS STATEMENT

This work uses public research benchmarks and does not involve human participants, user studies, or the collection of personal information. The proposed attack is studied for security evaluation, with the aim of identifying weaknesses in existing 3DGS watermarking systems and supporting more reliable ownership protection.

REPRODUCIBILITY STATEMENT

Section 3 specifies the threat model, target construction, and optimization procedure. Section 4.1 describes the datasets, baselines, and evaluation metrics. Appendix A provides implementation details, hyperparameters, and random seeds, while Appendix C specifies the supplementary owner verification protocol.

REFERENCES

Haonan An, Guang Hua, Hangcheng Cao, Zhengru Fang, Guowen Xu, Susanto Rahardja, and Yuguang Fang. Removing box-free watermarks for image-to-image models via query-based reverse engineering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19615–19622, 2026a.

Haonan An, Guang Hua, Zhiping Lin, and Yuguang Fang. Box-free model watermarks are prone to black-box removal attacks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(9):10610–10622, 2026b.

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*, pp. 5460–5469, 2022.

Mingshu Cai, Jiajun Li, Osamu Yoshie, Yuya Ieiri, and Yixuan Li. Where, What, Why: Toward explainable 3D-GS watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20701–20710, 2026.

Zixuan Chen, Guangcong Wang, Jiahao Zhu, Jianhuang Lai, and Xiaohua Xie. Guardsplat: Efficient and robust watermarking for 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16325–16335, 2025.

Yijia Guo, Wenkai Huang, Yang Li, Gaolei Li, Hang Zhang, Liwen Hu, Jianhua Li, Tiejun Huang, and Lei Ma. Splats in splats: Embedding invisible 3D watermark within gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Yuepeng Hu, Zhengyuan Jiang, Moyang Guo, and Neil Gong. A transfer attack to image watermarks. In *International Conference on Learning Representations*, pp. 18101–18127, 2025.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4733–4742, 2019.

Wenkai Huang, Yijia Guo, Gaolei Li, Lei Ma, Hang Zhang, Liwen Hu, Jiazheng Wang, Jianhua Li, and Tiejun Huang. Can protective watermarking safeguard the copyright of 3D gaussian splatting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5149–5157, 2026.

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 *Advances in Neural Information Processing Systems*, pp. 33037–33060, 2024.

Xiufeng Huang, Ziyuan Luo, Qi Song, Ruofei Wang, and Renjie Wan. Marksplatter: Generalizable watermarking for 3d gaussian splatting model via splatter image structure. In *ACM International Conference on Multimedia*, pp. 12189–12198, 2025.

Sumin In, Youngdong Jang, Utae Jeong, MinHyuk Jang, Hyeongcheol Park, Eunbyung Park, and Sangpil Kim. CompMarkGS: Robust watermarking for compressed 3D gaussian splatting. In *International Conference on Learning Representations*, 2026.

Youngdong Jang, Hyunje Park, Feng Yang, Heeju Ko, Euijin Choo, and Sangpil Kim. 3D-gsw: 3D gaussian splatting watermark for protecting copyrights in radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5938–5948, 2025.

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark-based detection of AI-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1168–1181, 2023.

Andre Kassis and Urs Hengartner. Unmarker: A universal attack on defensive image watermarking. In *IEEE Symposium on Security and Privacy (SP)*, pp. 2602–2620, 2025.

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.

Lijiang Li, Jinglu Wang, Xiang Ming, and Yan Lu. GS-Marker: Generalizable and robust watermarking for 3D gaussian splatting. *arXiv preprint arXiv:2503.18718*, 2025.

Nan Li, Yike Zeng, Qian Zhang, Qi Zhang, Zhiyi Pan, Wei Feng, and Liang Wan. Write Where It Matters: Policy-guided watermarks for 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6580–6590, 2026.

Yantao Lu, Yunhan Jia, Jianyu Wang, Bai Li, Weiheng Chai, Lawrence Carin, and Senem Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 940–949, 2020.

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph.*, 38(4):1–14, 2019.

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. *Commun. ACM*, 65(1):99–106, 2021.

Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20937–20946, 2025.

Hao Qin, Yukai Sun, Luyuan Chen, Mengxu Lu, Feng Zhang, Ming Kong, Zhenhong Du, and Qiang Zhu. Ngs-marker: robust native watermarking for 3D gaussian splatting. In *International Conference on Learning Representations*, pp. 70514–70539, 2026.

Huming Qiu, Zhaoxiang Wang, Mi Zhang, Xiaohan Zhang, Xiaoyu You, and Min Yang. The future unmarked: Watermark removal in AI-generated images via next-frame prediction. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124651–124676, 2026.

Mohit Singh and Weijun Xie. Approximation algorithms for D-optimal design. *Mathematics of Operations Research*, 45(4):1512–1534, 2020.

Boyu Wang, Jun Xia, and Mingsong Chen. Robust3DGSW: Toward robust watermarking for quantization-aware 3D gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19076–19084, 2026.

Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7639–7648, 2021.

Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? In *Advances in Neural Information Processing Systems*, volume 37, pp. 56644–56673, 2024.

Qingyuan Zeng, Shu Jiang, Jiajing Lin, Zhenzhong Wang, Kay Chen Tan, and Min Jiang. Fading the digital ink: A universal black-box attack framework for 3DGS watermarking systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38084–38092, 2026.

Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R. Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14993–15002, 2022.

Xuanyu Zhang, Jiarui Meng, Runyi Li, Zhipei Xu, Yongbing Zhang, and Jian Zhang. Gs-hider: Hiding messages into 3D gaussian splatting. In *Advances in Neural Information Processing Systems*, pp. 49780–49805, 2024.

Longjie Zhao, Ziming Hong, Zhenyang Ren, Runnan Chen, Mingming Gong, and Tongliang Liu. RDSplat: Robust watermarking against diffusion editing for 3D gaussian splatting. *arXiv preprint arXiv:2512.06774*, 2025.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8643–8672, 2024.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European Conference on Computer Vision*, pp. 682–697, 2018.

A Implementation Details

Table 4: Implementation settings. Target-construction parameters apply to AACT; optimization and SNSS settings are shared with CC and BI.

Parameter	Value
Rendering pool / pseudo-code estimation views	200 / 200
Carrier views $\lvert\Phi_c\rvert$ / stopping views $\lvert\Phi_p\rvert$	4 / 8
Strength bounds $(a_{\min}, a_{\max})$ / exponent β	$(0.25, 4)$ / 0.5
D-optimal coefficient γ	1
Trainable parameters	SH DC and rest
Adam learning rate / moment coefficients	0.004 / $(0.9, 0.99)$
Views per update / gradient norm cap	8 / 1
$(\lambda_{\text{tar}}, \lambda_{\text{vis}}, \lambda_{\text{reg}})$ in Eq. (8)	$(10, 2, 0.005)$
$(\omega_1, \omega_{\text{lp}}, \omega_{\text{ssim}}, \omega_{\text{hf}})$ in Eq. (12)	$(0.10, 0.60, 0.25, 0.02)$
Projection radii $(r_{\text{dc}}, r_{\text{rest}})$	$(0.45, 1.2)$
SNSS probe interval / reference intervals	5 / 3
Reference-rate fraction / consecutive low-rate checks	0.5 / 2
Minimum stopping iteration / maximum updates	15 / 80

A.1 Target Construction

We construct the carrier signatures in Eq. (4) from the strongest 0.5% of Gaussian responses for each bit and probe view, capped at 4096 Gaussians. The responses combine gradients with respect to SH DC and remaining SH coefficients. Signatures aggregate the normalized responses within this support, whereas flip strength uses the full gradient norm. The strength-weighting and D-optimal settings are listed in Table 4.

The balance constraint in Eq. (6) selects as nearly equal numbers of pseudo-zero and pseudo-one bits as possible. For $L = 48$, this gives 12 flips from each group whenever both groups contain at least 12 bits. Otherwise, we use the closest feasible allocation while retaining 24 flips in total. Starting from an empty set, we add the bit with the largest marginal log-determinant gain among candidates whose group quota is not exhausted. The selected set determines the target through Eq. (2). Appendix B illustrates this selection.

A.2 Optimization and Stopping

Fidelity and regularization. We specify the two additional terms in Eq. (8). Let $I_\phi = \mathcal{R}(\mathcal{G}, \phi)$ and $I_\phi^w = \mathcal{R}(\mathcal{G}^w, \phi)$ denote the current and original watermarked renders. Let P_d denote bilinear downsampling by a factor d followed by upsampling to the original resolution, and $H_d(I) = I - P_d(I)$. The visual loss is

$$\mathcal{L}_{\text{vis}} = \frac{1}{|\mathcal{V}|} \sum_{\phi \in \mathcal{V}} \big[\omega_1 \,\text{MAE}(I_\phi, I_\phi^w) + \omega_{\text{lp}} \,\text{MAE}(P_4(I_\phi), P_4(I_\phi^w)) + \omega_{\text{ssim}} \big(1 - \text{SSIM}(I_\phi, I_\phi^w)\big) + \omega_{\text{hf}} \,\text{SmoothL1}(H_8(I_\phi), H_8(I_\phi^w)) \big], \tag{12}$$

where MAE and SmoothL1 average over pixels and channels. The reference renders are fixed before optimization. For $q \in \{\text{dc}, \text{rest}\}$, the parameter regularizer is

$$\mathcal{L}_{\text{reg}} = \frac{1}{2} \sum_{q \in \{\text{dc},\text{rest}\}} \frac{1}{|\Theta_q^{\text{app}}|} \left\| \frac{\Theta_q^{\text{app}} - \Theta_q^{\text{app},0}}{\sigma_q} \right\|_F^2, \tag{13}$$

where $|\Theta_q^{\text{app}}|$ counts scalar entries and σ_q is the original tensor's standard deviation, lower-bounded by 10^{-6}. Loss weights and the relative radii used in Eq. (9) are given in Table 4.

SNSS. We evaluate Eq. (10) every five updates on eight fixed probe views. After the first crossing, the median of the first three complete post-crossing rates in Eq. (11) defines the fixed reference r_{ref}.

The interval ending at the first crossing is excluded. Once a positive reference is established, SNSS stops when two consecutive subsequent rates are at most $0.5r_{\mathrm{ref}}$, the current margin is positive, and at least 15 updates have been executed. Otherwise, optimization continues up to 80 updates. The margin in Eq. (10) uses the monitoring set $\mathcal{A}$. In the main text $\mathcal{A} = S^\star$ for AACT. Equivalently, $\mathcal{A} = \{\ell : m_\ell^t \neq \hat{m}_\ell\}$, fixed before optimization, which also applies to CC and BI. This includes all bit positions for BI and the positions differing from the constant target for CC. The set enters only the margin. Both optimization and probe losses use per-view full-target BCE. All three methods share the same probe views, stopping hyperparameters, and maximum iteration budget.

A.3 Experimental Configuration

Table 5 lists the 15 Blender and LLFF scenes evaluated in the main paper. Unless otherwise specified, we use seeds 0, 1, and 2 and a single fixed 48-bit RGB HiDDeN surrogate decoder (Zhu et al., 2018). CC, BI, and AACT share the appearance optimization and SNSS settings in Table 4, using per-view full-target BCE. CC uses a constant codeword opposite to the majority bit of the initial surrogate pseudo-code, with ties mapped to the all-zero codeword. BI uses the bitwise complement of the same pseudo-code. The carrier and stopping subsets are sampled once and held fixed. Metrics are averaged within each scene, equally across scenes, and then across seeds.

During attack optimization, the visual loss in Eq. (12) uses only original watermarked renders as image references. Dataset GT images are used exclusively for post-attack PSNR and SSIM evaluation.

Table 5: **Evaluated scenes in the main paper.** Our experiments cover 15 scenes across two datasets.

Dataset	Count	Scenes
Blender	8	Chair, Drums, Ficus, Hotdog, Lego, Materials, Mic, Ship.
LLFF	7	Fern, Flower, Fortress, Horns, Orchids, Room, Trex.

B An Illustrative Example of AACT

We illustrate the target construction in Sec. 3.2 with an eight-bit example. The example explains why carrier strength alone can select redundant bits, complementing the bit-selection ablation in Table 3(b). All values below are illustrative.

Balanced candidate selection. Consider the pseudo-code $\hat{\mathbf{m}} = (1, 0, 1, 1, 0, 0, 1, 0)$. Eq. (2) requires four flips. The balance constraint selects two bits from each pseudo-bit group. Table 6 gives strengths and simplified carrier signatures over four Gaussian coordinates. Bits 1 and 2 have high strengths but identical support. The other signatures expose candidates supported by different Gaussians.

Table 6: Illustrative strengths and carrier signatures. Bold bit indices identify the selected set $S^\star = \{1, 3, 5, 6\}$.

Bit ℓ	Pseudo-bit $\hat{m}_\ell$	Strength a_ℓ	Signature $\mathbf{c}_\ell$
1	1	4.0	$(1, 0, 0, 0)$
2	0	3.8	$(1, 0, 0, 0)$
3	1	3.0	$(0, 1, 0, 0)$
4	1	1.0	$(0, 1, 0, 0)$
5	0	2.8	$(0, 0, 1, 0)$
6	0	2.6	$(0, 0, 0, 1)$
7	1	0.8	$(0, 0, 1, 0)$
8	0	0.6	$(0, 0, 0, 1)$

Strength and complementarity. The median strength is $\bar{a} = 2.7$. Using the weights and kernel in Sec. 3.2, greedy selection first chooses bit 1. Its selection reduces the marginal log-determinant gain of bit 2 to $\log(1 + 3.8/(2.7 + 4.0)) = 0.4493$ because their signatures coincide. Bit 3 has disjoint

support and retains a larger gain, $\log(1+3.0/2.7) = 0.7472$, so it is selected next. With the pseudo-one quota filled, bits 5 and 6 complete the pseudo-zero quota. A strength-only rule would instead choose $\{1, 2, 3, 5\}$. Thus, the D-optimal criterion in Eq. (6) favors the weaker but complementary bit 6 over the redundant bit 2.

Target realization. Applying Eq. (2) gives

$$S^\star = \{1, 3, 5, 6\}, \qquad \mathbf{m}^t = (0, 0, 0, 1, 1, 1, 1, 0). \tag{14}$$

The target differs from $\hat{\mathbf{m}}$ in exactly four coordinates. All eight bits participate in the BCE of Eq. (7), including the four unselected bits whose target values are unchanged. Carrier signatures determine which bits to flip; appearance optimization still acts on the full set of Gaussian SH parameters.

Stopping. Suppose that the first three complete post-crossing rates in Eq. (11) are 0.20, 0.18, and 0.16. Their median gives $r_{\text{ref}} = 0.18$, so the saturation threshold is 0.09. Two subsequent consecutive rates of 0.07 and 0.06 trigger stopping once the current margin is positive and the minimum iteration requirement is met. These quantities describe the public optimization trajectory; owner-side disruption is measured separately after the attack.

C Watermark Recovery and Verification

We extend the target comparison in Table 2 by examining owner verification on individual and aggregated rendering views. GaussianMarker and GuardSplat are abbreviated as GM and GS, with the suffix indicating the owner message length.

Owner verification. Let K be the number of recovered bits matching an owner message of length L_o. We report the one-sided pass rate P_1, which accepts when $K \geq t_1$, and the inversion-aware pass rate P_2, which accepts when $\max(K, L_o - K) \geq t_2$. The latter tests whether verification remains possible after a bitwise inversion. Exact binomial thresholds under a uniform random-key null are chosen at a nominal false-positive rate of 10^{-3}: $(t_1, t_2) = (36, 36)$, $(26, 26)$, and $(45, 46)$ for 48-, 32-, and 64-bit owners, respectively. These common tests are used for comparison across systems. Owner-side measurements are performed only after the attack and do not guide target construction or stopping.

C.1 Novel-View Recovery and Rendering Fidelity

Table 7 compares CC, BI, and AACT on the 15 Blender/LLFF scenes in independent reruns, with watermark recovery evaluated on 64 randomly sampled novel views disjoint from the attack and calibration views. PSNR is measured on the native dataset GT test views. As specified in Appendix A.3, all attacks use the same public RGB HiDDeN surrogate and share appearance optimization and SNSS, so the comparison isolates target construction.

Table 7: **Novel-view recovery and rendering fidelity on Blender/LLFF.** Results are averaged over seeds 0, 1, and 2. Bold values indicate the best attack within each system; unattacked controls are excluded from ranking.

Metric	GM48				GS48			
	No attack	CC	BI	AACT	No attack	CC	BI	AACT
ΔBAR $\downarrow$	49.98	1.69	45.77	**0.00**	46.41	10.07	**8.65**	9.52
IDS $\uparrow$	0.03	96.28	8.30	**99.83**	7.15	79.87	**82.53**	80.32
$P_1 \downarrow$	100.00	**0.00**	**0.00**	**0.00**	100.00	6.15	**4.62**	5.52
$P_2 \downarrow$	100.00	**0.00**	95.80	**0.00**	100.00	6.15	**4.62**	5.52
PSNR $\uparrow$	27.07	25.62	25.39	**25.78**	32.40	28.57	27.89	**28.72**

On GM48, BI reduces P_1 to zero but retains $P_2 = 95.80\%$, reflecting predictable message inversion, whereas AACT achieves zero pass rates under both tests. On GS48, BI provides the strongest single-view disruption, while AACT improves over CC. AACT achieves the highest PSNR among the attacks in both systems. These results support the alignment dependence of endpoint targets and AACT's favorable disruption–fidelity trade-off.

C.2 MULTI-VIEW VERIFICATION

For multi-view verification, we average owner logits within disjoint groups of novel views before thresholding. Table 8 varies the group size from 1 to 64, testing whether aggregation can recover enough residual signal for verification.

Table 8: **Verification after multi-view aggregation on Blender/LLFF.** Column headings give the number of views whose owner logits are averaged before decoding. Results are pass rates (%) averaged over seeds 0, 1, and 2; lower is better.

System	Method	$P_1 \downarrow$						$P_2 \downarrow$
		1	4	8	16	32	64	64
GM48	No attack	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	CC	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BI	0.00	0.00	0.00	0.00	0.00	0.00	95.56
	AACT	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GS48	No attack	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	CC	6.15	9.17	10.83	11.67	13.33	13.33	13.33
	BI	4.62	8.19	11.39	13.89	16.67	20.00	20.00
	AACT	5.52	8.47	11.11	11.67	13.33	13.33	13.33

On GM48, AACT maintains zero one-sided pass rates across all group sizes and a zero inversion-aware pass rate at 64 views. On GS48, its P_1 increases from 5.52% to 13.33% at 64 views, matching CC and remaining below BI's 20.00%. Thus, aggregation can recover some residual signal, but BI's single-view advantage does not persist under larger view groups.

D GENERALIZATION

We examine whether the target-construction strategy in Sec. 3.2 remains effective on another dataset and when the owner and surrogate message lengths differ. Both evaluations retain the single fixed 48-bit RGB HiDDeN surrogate.

Generalization to Mip-NeRF 360. We evaluate GM48 on eight scenes from Mip-NeRF 360 (Barron et al., 2022): Bicycle, Bonsai, Counter, Garden, Kitchen, Room, Stump, and Treehill. Table 9(a) compares the attacks on novel views, and panel (b) reports seen- and novel-view recovery for AACT and the unattacked control. Results use seeds 0, 1, and 2.

Table 9: **Generalization to Mip-NeRF 360 with GM48.** Results cover eight scenes and three seeds. (a) Novel-view attack comparison; bold marks the best attack. (b) Seen- and novel-view recovery for the unattacked and AACT-attacked assets.

(a) Novel-view recovery and fidelity

Method	ΔBAR $\downarrow$	IDS $\uparrow$	$P_1 \downarrow$	$P_2 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
No attack	44.09	11.63	96.18	96.18	30.45	0.921
CC	1.84	92.89	**0.00**	**0.00**	27.24	**0.834**
BI	44.29	11.22	**0.00**	98.16	26.81	0.820
AACT	**1.71**	**94.63**	**0.00**	**0.00**	**27.25**	**0.834**

(b) Seen- and novel-view recovery

Method	Seen			Novel		
	ΔBAR $\downarrow$	$P_1 \downarrow$	Alignment	ΔBAR $\downarrow$	$P_1 \downarrow$	Alignment
No attack	50.00	100.00	100.00	44.09	96.18	100.00
AACT	0.23	0.08	100.00	1.71	0.00	100.00

AACT achieves $\Delta\text{BAR} = 1.71\%$ and $\text{IDS} = 94.63\%$, with zero pass rates under both verification tests. BI again reduces P_1 to zero while retaining $P_2 = 98.16\%$, consistent with the highly aligned GM48 results on Blender/LLFF. AACT also has the highest PSNR and ties CC for the highest SSIM

among the attacks. The view comparison in panel (b) reports a near-zero seen-view pass rate and zero novel-view passes. Together, these results extend the observed disruption–fidelity trade-off beyond Blender and LLFF while retaining the same surrogate.

Generalization to different owner message lengths. We evaluate RGB HiDDeN owners with 32- or 64-bit messages, denoted as GS32, GM32, and GM64, while keeping the public surrogate fixed at 48 bits. CC, BI, and AACT use the same appearance optimization and SNSS settings. Table 10 reports results averaged over seeds 0, 1, and 2 on the 15 Blender/LLFF scenes.

Table 10: **Generalization to different owner message lengths.** RGB HiDDeN owners encode 32- or 64-bit messages; all attacks use one fixed 48-bit RGB HiDDeN surrogate. Results are averaged over 15 Blender/LLFF scenes and three seeds (0, 1, and 2). **Bold** marks the best attack within each owner setting; unattacked controls are excluded from ranking.

Metric	GS32				GM32				GM64			
	No attack	CC	BI	AACT	No attack	CC	BI	AACT	No attack	CC	BI	AACT
ΔBAR $\downarrow$	48.85	7.58	**6.92**	6.96	43.25	21.07	**16.06**	16.14	40.64	18.75	**11.10**	11.23
IDS $\uparrow$	2.28	84.74	**85.44**	85.18	13.49	57.67	**67.94**	67.88	18.63	65.26	**78.03**	77.60
P_1 $\downarrow$	100.00	**0.00**	**0.00**	**0.00**	86.46	11.46	6.98	**3.85**	89.38	31.15	23.12	**18.44**
P_2 $\downarrow$	100.00	**0.00**	**0.00**	**0.00**	86.46	11.46	6.98	**3.85**	89.06	28.12	19.58	**17.19**
PSNR $\uparrow$	32.85	**25.32**	25.10	25.31	27.10	23.32	23.19	**23.47**	27.88	23.92	23.83	**23.95**
SSIM $\uparrow$	0.967	0.833	0.825	**0.835**	0.911	0.785	0.779	**0.790**	0.914	0.788	0.783	**0.792**

BI achieves lower ΔBAR and higher IDS than CC in all three settings. This ordering differs from the highly aligned GM48 results, supporting the main observation that endpoint-target effectiveness depends on the surrogate–owner relationship. AACT remains close to the best disruption among the compared attacks: its ΔBAR exceeds BI by only 0.04, 0.08, and 0.13 percentage points on GS32, GM32, and GM64, respectively. It also achieves lower verification rates on GM32 and GM64, and higher PSNR and SSIM than BI in all three settings. Because the half-Hamming target is defined in the surrogate's own code space, its construction does not require equal owner and surrogate message lengths. These results support its effectiveness under the evaluated length mismatches without selecting an endpoint for each owner.

E Qualitative Results and Limitations

Blender scenes. Figure 6 extends the qualitative comparison in Figure 3 to all eight Blender scenes. AACT better preserves object colors and fine appearance details than GMEA, as seen in the drum surfaces, microphone grille, and Lego textures. These observations are consistent with the fidelity advantage in Table 1.



Figure 6: **Qualitative comparison on Blender.** All eight scenes are shown for GuardSplat (left) and GaussianMarker (right). Within each system, columns show the original watermarked render (WM Input), GMEA, and AACT (Ours) at the same view.

LLFF scenes. Figure 7 presents the seven LLFF scenes. AACT better retains leaf and flower detail in Fern, Flower, and Orchids, while GMEA introduces more pronounced color variations and texture degradation.

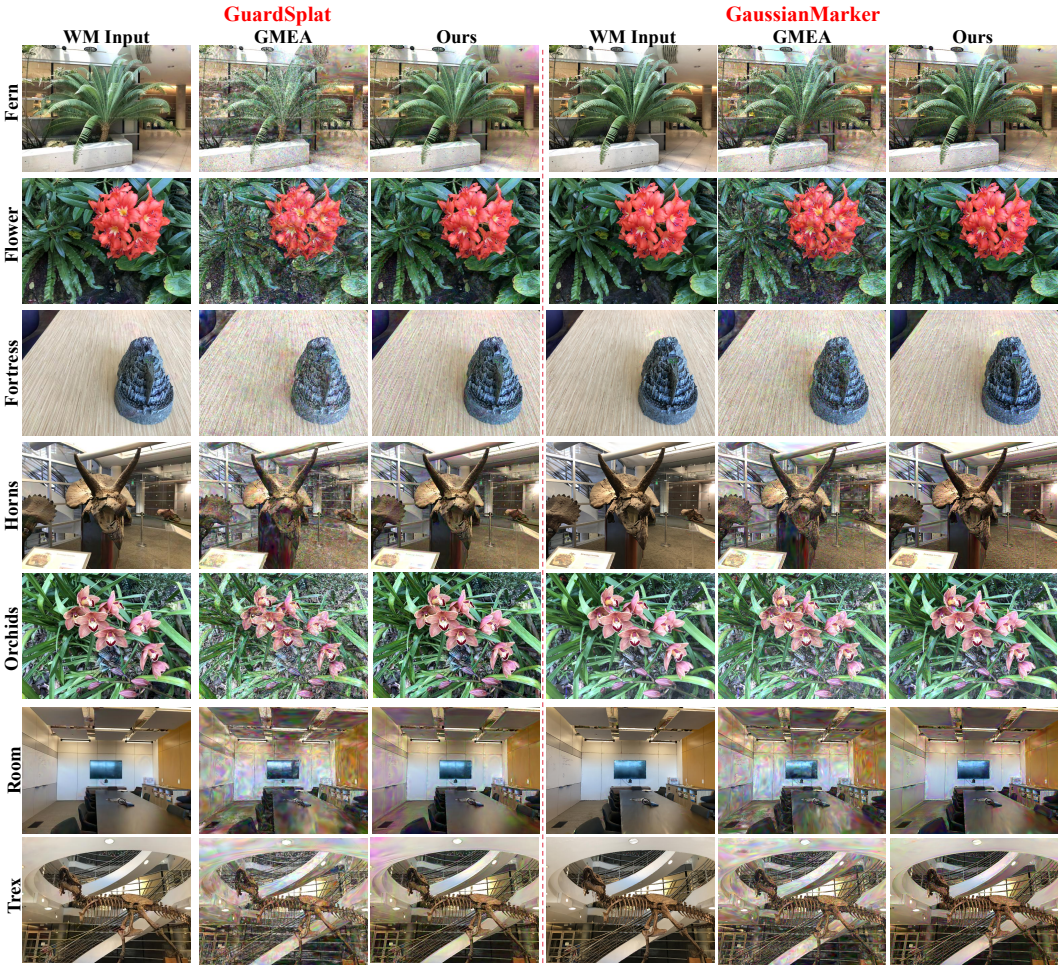


Figure 7: **Qualitative comparison on LLFF.** The seven scenes use the same system grouping and method order as Figure 6. Each comparison uses a fixed view.

Limitation. Figure 8 illustrates a limitation on the challenging Room scene. Although AACT achieves higher full-view PSNR than GMEA, local color artifacts remain visible in the enlarged cabinet and wall regions. AACT preserves Gaussian geometry and the complete primitive set, providing a basis for subsequent appearance fine-tuning. Exploring such refinement to improve local rendering quality while maintaining watermark disruption is a direction for future work.

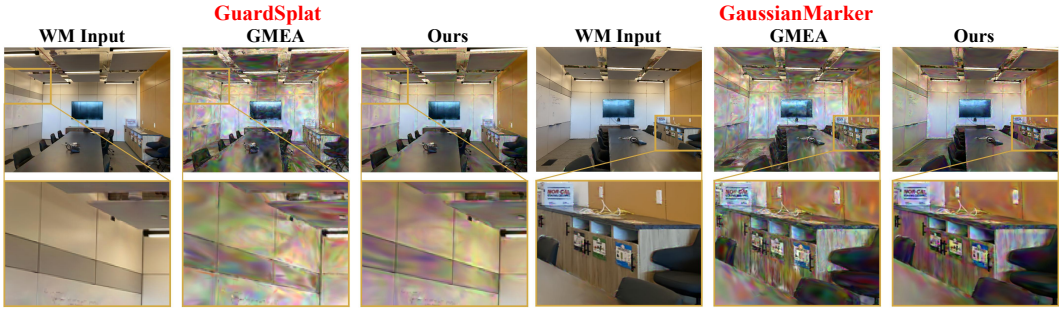


Figure 8: **Limitation on the challenging Room scene.** Within each system, the top row shows one held-out view and the bottom row enlarges a fixed region for comparing local color artifacts.