DEGauss: Defending Against Malicious 3D Editing for Gaussian Splatting

Lingzhuang Meng[1], Mingwen Shao[2]*, Yuanjian Qiao[3], Xiang Lv[1]
[1] Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software,
Qingdao Institute of Software, College of Computer Science and Technology,
China University of Petroleum (East China), China
[2] Artificial Intelligence Research Institute, Shenzhen University of Advanced Technology, China
[3] College of Computer Science (College of Software), Inner Mongolia University, China
lzhmeng1688@163.com, smw278@126.com, yjqiao58@163.com, lvxiang1997@126.com

Abstract

3D editing with Gaussian splatting is exciting in creating realistic content, but it also poses abuse risks for generating malicious 3D content. Existing 2D defense approaches mainly focus on adding perturbations to single image to resist malicious image editing. However, there remain two limitations when applied directly to 3D scenes: (1) These methods fail to reflect 3D spatial correlations, thus protecting ineffectively under multiple viewpoints. (2) Such pixel-level perturbation is easily eliminated during the iterations of 3D editing, leading to failure of protection. To address the above issues, we propose a novel **D**efense framework against malicious 3D **E**diting for **Gauss**ian splatting (**DEGauss**) for robustly disrupting the trajectory of 3D editing in multi-views. Specifically, to enable the effectiveness of perturbation across various views, we devise a view-focal gradient fusion mechanism that dynamically emphasizes the contributions of the most challenging views to adaptively optimize 3D perturbations. Furthermore, we design a dual discrepancy optimization strategy that both maximize the semantic deviation and the edit direction deviation of the guidance conditions to stably disrupt the editing trajectory. Benefiting from the collaborative designs, our method achieves effective resistance to 3D editing from various views while preserving photorealistic rendering quality. Extensive experiments demonstrate that our DEGauss not only performs excellent defense in different scenes, but also exhibits strong generalization across various state-of-the-art 3D editing pipelines.

1 Introduction

Recent advances in scene editing with 3D Gaussian Splatting (3DGS) [17] have substantially enhanced the controllability and expressiveness scene manipulation driven by natural language prompts, largely powered by the integration of diffusion models with neural 3D representations [12, 39]. However, such convenience also brings significant security risks: anyone with access to a rendered 3D model can change identity, appearance, or contextual details without authorization [25, 22], which can lead to serious consequences such as identity deception, misinformation, or reputational damage [31, 40]. Therefore, it is imperative to develop effective protection method for 3D digital assets and personal portraits against unauthorized modifications, in order to prevent the spread of malicious content.

Unlike 2D editing that operates on single image [15, 38], 3D editing fundamentally involves manipulating spatially structured data [20, 35], which brings in unique properties such as multi-view

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

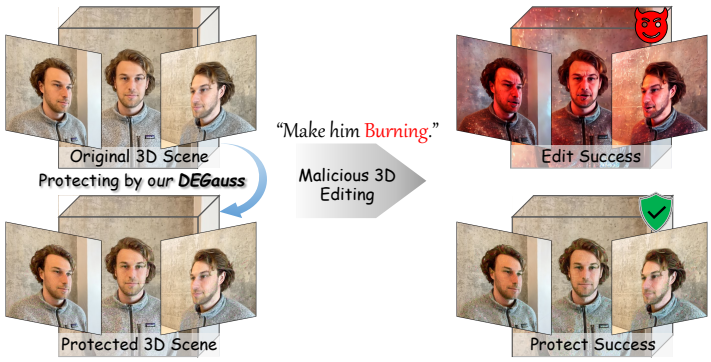

Figure 1: Malicious 3D editing and our defense. The 3D scenes protected by our DEGauss are able to disrupt the direction of malicious editing and preserve the original 3D appearance and structure.

consistency, structural integrity, and global iterative optimization. Existing studies [24, 3] on 3D editing based on Gaussian splatting and diffusion models have emphasized better semantic controllability, multi-view consistency, and editing flexibility. These methods aim to enable precise modifications of 3D content that align with user intent by local region mapping [7], while ensuring consistency across diverse viewpoints through the integration of semantic information from different viewpoints [4, 34]. Furthermore, recent works [18, 19] have increasingly focused on the editing efficiency and speed, enabling high-quality results with reduced computational cost and minimal guidance. Despite these significant advances, existing researches neglect the security risks posed by malicious use of 3D editing, which raises concerns about unauthorized modifications.

To defend against malicious editing on 2D images, various adversarial-based defense methods [22, 14, 26] have been proposed to disrupt the editing capabilities of diffusion models through embedded imperceptible perturbations. For example, AdvDM [22], Mist [21] and PhotoGuard [30] optimize the perturbations by maximizing the noise prediction loss of the diffusion model, thereby inducing significant degradation in downstream editing results. Posterior Collapse Attack [10] achieves attacks in latent space by inducing a posterior collapse in the VAE encoder, thus disrupting a variety of editing methods based on latent diffusion models. In addition, Semantic Attack [23] and AdvPaint [16] degrade editing quality by manipulating the self-attention and cross-attention mechanisms within the diffusion process, leading to misaligned attention maps and impaired semantic alignment. However, the aforementioned 2D defense methods are inherently unsuitable for 3D scenarios due to two main limitations: (1) Their 2D perturbations failed to reflect 3D spatial correlations, leading to poor protection across varying viewpoints. (2) Such perturbations are also easily erased during iterative optimization of 3D editing, resulting in defense failure.

To tackle the above challenges, we propose a tailored **D**efense framework against malicious 3D **E**diting for **Gauss**ian splatting (**DEGauss**), which stably disrupts the multi-view editing trajectory to suppress unauthorized 3D modifications, as shown in Fig. 1. Specifically, to ensure the effectiveness of perturbations across different viewpoints, we design a view-focal gradient fusion mechanism that dynamically emphasizes the gradient of the most challenging viewpoints in broadly sampled cameras, thereby adaptively optimizing the 3D perturbations under varying viewpoints. Furthermore, we devise a dual discrepancy optimization strategy that simultaneously maximizes the semantic deviation and the directional bias of the guidance condition, therefore enabling the 3D perturbation to robustly disrupt editing trajectories throughout the iterative optimization. With the support of these two dedicated designs, our method achieves strong resistance to multi-view editing, while preserving high-fidelity rendering with photorealistic visual details. Extensive experiments illustrate that our DEGauss effectively prevents malicious 3D editing across diverse scenes and exhibits promising generalization across a variety of 3D editing frameworks.

The main contributions of this paper are as follows:

- We propose DEGauss, a noval defense framework against malicious 3D editing for Gaussian splatting that actively disturbs the editing trajectory from multiple viewpoints. To the best of our knowledge, this is the first framework for defense against 3D malicious editing.

- A view-focus gradient fusion mechanism is designed to adaptively update 3D perturbations by emphasizing challenging views, thereby enhancing defense effectiveness in multi-views.
- We devise a dual-discrepancy optimization strategy that amplifies semantic divergence and editing directions errors in iterative optimization, stably disrupting the editing trajectory.
- Experiments on several datasets and state-of-the-art 3DGS editing schemes demonstrate the effectiveness and generalization of our DEGauss.

2 Related Works

3D Editing with Gaussian Splatting. Recent researches on 3D editing with 3DGS have focused almost on improving the fidelity [4, 18, 33], controllability [39, 7, 12, 28], and efficiency [18, 19]. For instance, to achieve controllable editing, GaussianEditor [7] introduces a dynamic semantic tracking strategy for precise localization by back-projecting 2D segmentation masks into 3D Gaussians. To ensure multi-view consistency, DGE [4] integrates spatio-temporal self-attention and epipolar constraints to achieve effective fusion of edits across views, while GaussCtrl [34] leverages geometric constraints on depth maps and cross-view attention mechanism to align edits across different views. To improve editing efficiency, ProEdit [3] adopts subtask-based optimization to gradually realize complex edits with better stability. On the contrary, DreamCatalyst [18] accelerates convergence via improved inversion and loss design, while EditSplat [19] further enhances speed through attention-weighted pruning and hierarchical densification. Despite the impressive progression of 3D editing, there are also raised concerns about its potential misuse to maliciously modify private 3D assets. Aiming at this problem, we propose DEGauss, the first defense framework against malicious 3D editing for protecting digital assets from arbitrary manipulation by unauthorized editor.

Defenses against Malicious Image Editing. To defend against malicious images editing, a series of adversarial-based defense methods [22, 6, 5, 14, 26] have been proposed to disrupt the editing process by exploiting imperceptible pixel-space perturbations. Among them, AdvDM [22] introduces a systematic theoretical framework that maximizes denoising loss of the diffusion models [29, 13], thereby generating perturbations to hinder feature extraction. On this basis, Mist [21] further combines semantic and texture losses to improve transferability of the perturbations, while DiffusionGuard [8] prevents the synthesis in sensitive regions by interfering with the early stages of the denoising process. In addition, SDS [36] accelerates optimization and reduces consumption through score distillation sampling, while Posterior Collapse Attack [10] induces potential spatial collapse by perturbing the VAE encoder in the latent diffusion model, thereby significantly disrupting the semantics of the encoding. Recently, Semantic Attack [23] and AdvPaint [16] perturb the text-image cross-attention or image self-attention in the diffusion model to distract the model attention to the specific region, and both of them showing better performance than conventional approaches. These defense schemes are investigated from the output space to the model structure to construct a more defense system against malicious editing of 2D images.

However, the above methods pose intrinsic challenges when applied directly to 3D scene protection, as they are specifically designed for single-view and single-pass 2D editing. On the one hand, the perturbations generated by 2D defense are unable to reflect 3D spatial relationships, leading to insufficient effectiveness across multiple viewpoints. On the other hand, simple perturbations optimized for single-pass editing are typically eliminated during the iterative optimization in 3D editing, resulting in failure of protection. In contrast, we propose DEGauss, a specialized defense framework against malicious 3D editing, which combines view-focal gradient fusion with dual-discrepancy optimization to robustly disrupt multi-view editing results during iterations of 3D editing.

3 Methodology

3.1 Preliminary

3D Editing with Gaussian Splatting. Existing mainstream of 3D editing with 3DGS employ pre-trained 2D diffusion models (e.g., InstructPix2Pix [2], ControlNet [37]) as powerful generative priors to supervise the editing results. Within these framework, text prompts are used to guide diffusion-based image editing for multiple viewpoints. The resulting edited images serve as supervision to update the Gaussian parameters $\Theta = \{\mu, \Sigma, \alpha, c\}$, where μ denotes the 3D center position, Σ

represents the covariance matrix controlling shape, α is the opacity, and c defines the RGB color. The updated 3D representation is iteratively rendered to generate a new image for further editing.

Formally, given a constructed scene G, the goal of the editor is to update the Gaussian parameters Θ so that its rendered images from all viewpoints match the edited images according to the text prompt y_{text}. This update is performed iteratively:

$$\Theta = \arg\min_{\Theta} \mathbb{E}_{k\in\mathcal{U}(0,K),v\in V}\mathcal{L}_{edit}(I_v^k, D_\phi(I_v^k, I_v, y_{\text{text}})), \tag{1}$$

where $I_v^k = R_v(G^k)$ denotes the k-th round of iterations and the v-th view of the rendered image, I_v represents the initial rendered image being used as image guidance to preserve the original information. D_ϕ denotes the diffusion model, and $\mathcal{L}_{edit}$ is the loss of rendering, which is commonly used to align the appearance of rendering (e.g., L_1, LPIPS) or geometric depth.

Defense against Malicious Image Editing. To defend against malicious image editing, existing researches has proposed adversarial-based methods to disrupt the editing process of diffusion models. The core principle of these approaches is to introduce a small-magnitude perturbation δ to the input image x to make the diffusion model deviate from the original denoising process, thus affecting the normal editing operation. It can be formalized as:

$$\delta := \arg\min_{\delta} p(x+\delta), \quad s.t.\|\delta\| < \xi, \tag{2}$$

where $p(x)$ denotes the probability distribution of the image x generated by the diffusion model and ξ denotes the bound limit of the perturbation. To solve Eq. (2), AdvDM [22] represents the solution of the defense perturbation δ uniformly according to Monte Carlo as follows:

$$\delta = \arg\max_{\|\delta\|<\xi} \mathbb{E}_{t\sim\mathcal{U}(1,T),\epsilon\sim\mathcal{N}(0,\mathbf{I})} \left\|\epsilon_\theta(x_t, x+\delta, y_{\text{text}}, t) - \epsilon\right\|_2^2, \tag{3}$$

where $t \sim \mathcal{U}(1,T)$ denotes the number of denoising steps in the diffusion model, $\epsilon \sim \mathcal{N}(0,\mathbf{I})$ denotes normally distributed noise, and ϵ_θ is the denoising network. By encouraging the noise predictions to deviate from the correct values at all time steps t, the denoising results of the diffusion model can be actively perturbed and ultimately achieve a robust defense against malicious image editing.

3.2 Problem Definition

In this paper, we present a defense against malicious 3D editing, aiming to prevent unauthorized modifications to 3D scenes performed by diffusion model-based editors. For a scene G constructed by 3DGS, let $p_\phi(G_{\text{tar}}|G, y_{\text{text}})$ is the conditional probability distribution of the target 3D scene G_{tar} produced by the diffusion editor (parameterized by ϕ) given a text prompt T. We define the defense against malicious 3D editing as a probabilistic optimization problem.

Definition 3.1 (Defense against malicious 3D editing). Given a 3D editor, there exists 3D perturbation Δ added to the original scene G that minimize the conditional probability of generating a valid edit result at text prompt y_{text}. The perturbation Δ is given by the following equation:

$$\Delta := \arg\min_{\Delta} p_\phi(G_{\text{tar}}|G+\Delta, y_{\text{text}}), \quad s.t. \quad \|\Delta\| < \xi. \tag{4}$$

Our goal is not limited to dissimilarity to a specific target distribution G_{tar}, but to reduce the probability of similarity to any edit distribution. Thus, the above equation can be written as:

$$\Delta := \arg\max_{\|\Delta\|<\xi} \mathbb{E}_{G_{\text{tar}}\sim p_\phi(\cdot|G,y_{\text{text}})}[-\log p_\phi(G_{\text{tar}}|G+\Delta, y_{\text{text}})]. \tag{5}$$

Combined with the optimization objective for 3D editing given in Eq. (1), we can reinterpret the defense problem as a maximization of the editing loss, as follows:

$$\Delta := \arg\max_{\|\Delta\|<\xi} \mathbb{E}_{k,v}\, \mathcal{L}_{\text{edit}}(I_v^k, D_\phi(I_v^k, R_v(G+\Delta), y_{\text{text}})). \tag{6}$$

However, this rendering loss is essentially an iterative fitting process to the explicit model, aiming to align the rendered image with the target edited image produced by the diffusion model. It is not possible to optimize the perturbation by maximizing this render loss $\mathcal{L}_{\text{edit}}$. Therefore, we turn to replace the original editing loss with a differentiable diffusion loss, which drives the 3D edit to fit to

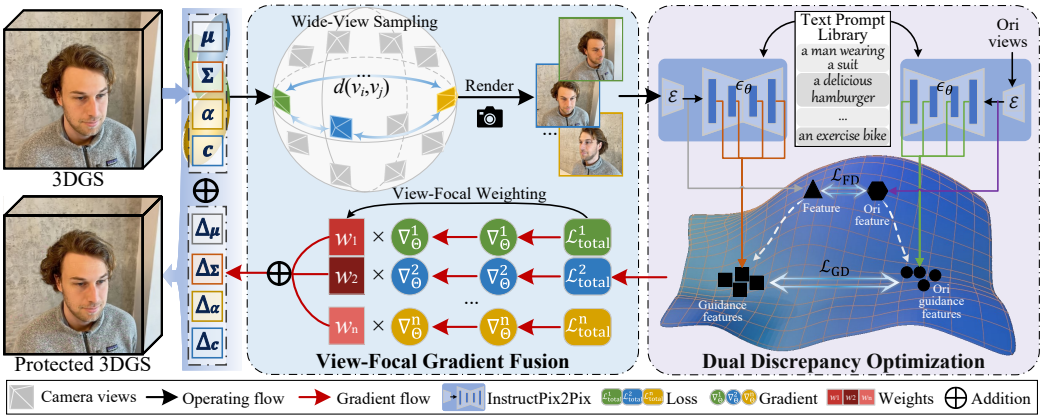

Figure 2: Overview of the proposed DEGauss. Firstly, a wide-viewpoint sampling is performed to obtain a rendered image of the 3D scene, and then feature discrepancy loss and guidance discrepancy loss are computed between the rendered and original view images. These losses from multiple viewpoints are subsequently fused using a view-focal weighting to guide perturbation optimization. The perturbation $\Delta = \{\Delta_\mu, \Delta_\epsilon, \Delta_\alpha, \Delta_c\}$ is initialized to zero and optimized in iterations.

the wrong edit direction by making the diffusion process in multiple views deviate from the original edit direction. This objective is formally defined as follows:

$$\min_{\|\Delta\|<\xi} \mathbb{E}_{v\in V} \mathcal{L}_{\text{diff}}(\mathcal{R}_v(G+\Delta), y_{\text{text}}) \tag{7}$$

$$= \max_{\|\Delta\|<\xi} \mathbb{E}_{k,v,t,\epsilon} \left\|\epsilon_\theta(I_v^k, \mathcal{R}_v(G+\Delta), y_{\text{text}}, t) - \epsilon\right\|_2^2. \tag{8}$$

Although Eq. (8) constructs theoretically reasonable attack targets based on the noise prediction of the diffusion model, it results in limited direct optimization due to the fact that the denoiser of the diffusion model is extremely robust to perturbations and the gradient is weak and unstable [36]. Therefore, we propose two loss terms, Feature Discrepancy (FD) and Guidance Discrepancy (GD), which directly measure the semantic differences and bias in the guidance direction in latent space. These contribute to more stable convergence and better gradient propagation.

$$\begin{aligned}\Delta := \arg\max_{\|\Delta\|<\xi} \mathbb{E}_{k,v,t,\epsilon} & \underbrace{\left\|\varepsilon(\mathcal{R}_v(G+\Delta)) - \varepsilon(I_v)\right\|_2^2}_{\text{Feature Discrepancy}} \\ & + \underbrace{\left\|\epsilon_\theta(I_v^k, \mathcal{R}_v(G+\Delta), y_{\text{text}}, t) - \epsilon_\theta(I_v', I_v, y_{\text{text}}, t)\right\|_2^2}_{\text{Guidance Discrepancy}},\end{aligned} \tag{9}$$

where ε represents the image encoder. The purpose of FD is to drive the protected view away from the original feature in latent space and increase the uncertainty of the editor. While the purpose of the GD is to alter the guidance direction of the diffusion model to deviate from the original trajectory, weakening its semantic understanding of y_{text} and editing behavior.

Following the above analysis, we propose DEGauss, a dedicated framework that introduces targeted perturbations to disrupt editing across multiple views, as illustrated in Fig. 2. Our DEGauss consists of two key components: a view-focal gradient fusion mechanism and a dual-discrepancy optimization strategy. These components work collaboratively to optimize 3D spatial perturbations that reliably disrupt multi-view editing throughout the iterative process.

3.3 Dual Discrepancy Optimization

Based on Eq. (9), we devise a dual discrepancy optimization strategy that provides supervision from both the semantic feature and guidance direction in the latent space. Specifically, this module contains two dedicated loss functions: feature discrepancy loss and guidance discrepancy loss, which jointly guide the optimization of the perturbations, thus deviating editing results from target semantics.

Feature Discrepancy Loss. The feature discrepancy loss is used to measure the consistency between the original views and the perturbed 3D views in the latent space, and we utilize the encoder of pre-trained InstructPix2Pix [2] model to extract the feature representations of the rendered image and the original image, which can be represented as:

$$\mathcal{L}_{\text{FD}} = -\mathbb{E}_{v\in V} \left\| \varepsilon \left(\mathcal{R}_v(G+\Delta) - \varepsilon(I_v) \right) \right\|_2^2 . \tag{10}$$

Guidance Discrepancy Loss. The guidance discrepancy loss is designed to ensure that the editing direction of the protected 3D scene deviates from that of the original scene under the same viewpoint. To achieve this, we utilize features from the noise space of the pre-trained InstructPix2Pix [2] model to represent guidance features. The loss is defined as follows:

$$\mathcal{L}_{\text{GD}} = -\mathbb{E}_{k,v,t,\epsilon,y_{\text{text}}\in Y_{\text{lib}}} \left\| \epsilon_\theta(I_v^k, \mathcal{R}_v(G+\Delta), y_{\text{text}}, t) - \epsilon_\theta(I'_v, I_v, y_{\text{text}}, t) \right\|_2^2 , \tag{11}$$

where Y_{lib} represent a library of text prompt, as in DreamFusion [27], which contains 415 different prompts. It is used to introduce semantic diversity and prevent overfitting to a specific prompt.

Overall Loss. During training, the above two losses are jointly optimized with the main rendering loss, and the total loss can be expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{render}} + \lambda_{\text{FD}} \cdot \mathcal{L}_{\text{FD}} + \lambda_{\text{GD}} \cdot \mathcal{L}_{\text{GD}}, \tag{12}$$

where $\mathcal{L}_{\text{render}}$ denotes the rendering loss, defined as the L_2 distance between rendered views and original views. λ_{FD} and λ_{GD} are the hyperparameters for the feature and guidance terms, respectively. The total loss is then used for view-focal gradient fusion to update the perturbation parameters.

3.4 View-Focal Gradient Fusion

During 3D object optimization, updates from different viewpoint may interfere with one another, leading to inconsistencies across multiple viewpoints. This effect is particularly noticeable in the optimization of fine-grained perturbations, resulting in extremely poor protection in some viewpoints. To mitigate this issue, we devise a View-Focal Gradient Fusion (VFGF) strategy that consists of two parts, a wide-view sampling and a view-focal weighting, which aims to improve the consistency and effectiveness of the perturbations across multiple views.

Wide-View Sampling. To ensure a uniform spatial distribution and broad coverage of the selected viewpoints, we design the wide-view sampling strategy to avoid concentration in locally similar or redundant views. For the a reference viewpoint v_i and v_j, each is associated with a camera translation vector $\mathbf{T} \in \mathbb{R}^3$ and the rotation matrix $\mathbf{R} \in \text{SO}(3)$. We define the distance between two viewpoints as the sum of normalized translation and angular differences:

$$d(v_i, v_j) = \underbrace{\frac{\|\mathbf{T}_i - \mathbf{T}_j\|_2}{D_{\text{max}}}}_{\text{Translation distance}} + \underbrace{\frac{\angle(\mathbf{R}_i, \mathbf{R}_j)}{\pi}}_{\text{Rotation angle}}, \tag{13}$$

$$\angle(\mathbf{R}_i, \mathbf{R}_j) = \arccos\left(\frac{trace(\mathbf{R}_j \mathbf{R}_i^\top) - 1}{2} \right), \tag{14}$$

where D_{max} is the maximum translation distance in view space and $trace(\cdot)$ denotes the trace of the matrix. Both components are mapped to $[0, 1]$ and equally weighted. Start from a initial set of viewpoints $V_1 = \{v_{\text{ref}}\}$, we select the next viewpoint furthest from all previously selected viewpoints based on all the statistics until N viewpoints have been selected:

$$V_N = V_{N-1} \cup \left\{ v^* \;\middle|\; v^* = \arg \max_{v_j \in \mathcal{V} \setminus V_{N-1}} \min_{v_i \in V_{N-1}} d(v_i, v_j) \right\}. \tag{15}$$

At each iteration, from the unselected viewpoints, find that viewpoint that has the largest distance from the nearest point in the current sampling set and add it to the sampling set. This sampling strategy avoids focusing the samples on locally similar viewpoints, thus providing stronger global structural constraints for the subsequent gradient fusion.

View-Focal Weighting. For the widely sampled set of viewpoints, we further propose a view-focal weighting strategy, which weights the gradient according to the difficulty of each viewpoint for

subsequent perturbation updates. Specifically, we treat the loss associated with each viewpoint as an indicator of its resistance to editing, and assign weights to the viewpoints based on the magnitude of the loss value, formally defined as follows:

$$\nabla_\Theta \mathcal{L}_{\text{focal}} = \sum_{v \in V} w_v \cdot \nabla_\Theta \mathcal{L}^v_{\text{total}}, \quad \text{where} \quad w_v = \frac{(\mathcal{L}^v_{\text{total}} + \tau)^\gamma}{\sum_{v' \in V} (\mathcal{L}^{v'}_{\text{total}} + \tau)^\gamma}, \tag{16}$$

where $\mathcal{L}^v_{\text{total}}$ denotes the total loss at viewpoint v, γ is focusing parameter controlling the degree of nonlinearity in the weighting, and τ is a stabilization term to prevent zero values. This strategy biases the optimizer toward viewpoints with higher loss values (i.e., with weak defenses against editing), thereby generating perturbations that are effective across views to improve resistance to editing.

Benefiting from the above designs, our DEGauss effectively optimize 3D perturbations in space while maintaining perturbation validity across different views. This enables stable and view-generalizable protection against malicious 3D editing throughout the iterative optimization process.

4 Experiments

4.1 Experimental Setup

Datasets and Editing Models. We verified the effectiveness on common 3D editing dataset [11, 32, 1], including 'face', 'girl', 'person', 'bear', 'bicycle', and 'garden' scenes, with varying viewpoints and data scales. In addition, we validate our generalization leveraging the latest released 3D editing models, including GaussianEditor [7], DGE [4], DreamCatalyst [18], and EditSplat [19].

Baselines. Since the defense schemes for malicious 3D editing are still pending, we compare our DEGauss with state-of-the-art 2D editing defense schemes and migrated them to 3DGS to ensure fairness, including AdvDM [22], Mist [21], SDS [36], and AdvPaint [16].

Evaluation Metrics. We utilize Peak Signal-to-Noise Ratio (PSNR) to measure the difference between protected and original samples, reflecting the imperceptibility of perturbations. In addition, Contrastive Language-Image Pretraining (CLIP) similarity evaluates the semantic gap between protected and normal editing results. CLIP-T measures the semantic distance between the protected result and the editing prompt. CLIP Direction similarity (CLIP-D) [9] quantifies the consistency between text differences (from source to editing) and image differences (from source to edited).

Implementation Details. All experiments are conducted on a single NVIDIA RTX 4090 GPU. We set the number of sampled views $N = 6$, the weighting factor $\tau = 1e\text{-}6$, and $\gamma = 1.0$. To balance loss terms, we set hyperparameters $\lambda_{\text{FD}} = \lambda_{\text{GD}} = 1e\text{-}5$. The total number of training steps is set to 2,000. More settings and algorithm are provided in the **Supplementary**.

4.2 Comparisons

We compare existing 2D-based editing defense methods with our DEGauss, focusing on the ability to resist malicious editing in 3D scenarios. As shown in Fig. 3, existing 2D defense methods suffer from obvious editing artifacts and multi-view inconsistencies when directly applied to 3D scenes. For example, AdvDM [22] and Mist [21] introduce some noise or artifacts that degrade the quality of the edits, but still maintain normal editing effects without preserving the original scene detail. SDS [36] exhibit weaker suppression of malicious edits, offering limited resistance. Although AdvPaint [16] performs slightly better in suppressing editing effects, it still introduces noticeable editing traces, noise, and inconsistencies across different viewpoints. In contrast, our DEGauss better preserves the global structure and appearance of the original scene, producing natural and artifact-free renderings with high visual fidelity across multiple views.

Moreover, Table 1 quantitatively evaluates our DEGauss against existing methods. It can be seen that our DEGauss achieves the highest PSNR, indicating that the introduced perturbations are more imperceptible. Regarding CLIP-based scores, our DEGauss attains the lowest similarity between the edited and original samples, demonstrating its effectiveness in disrupting malicious editing trajectories. In contrast, existing 2D defense methods such as AdvDM and Mist exhibit significantly lower performance in all metrics. Even AdvPaint, which performs relatively better among the baselines, still lags behind DEGauss on several metrics. Both visual and quantitative comparisons

Figure 3: Visual comparison with existing methods. Comparisons include AdvDM [22], Mist [21], SDS [36], and AdvPaint [16]. Our DEGauss is resistant to 3D editing achieves optimal visualization in preserving the original 3D scene. The editing method used is GaussianEditor.

Table 1: Quantitative comparison with existing methods. Our DEGauss achieves the superior PSNR while offering the optimal resistance to 3D editing. The editing method used is GaussianEditor. Red indicates optimal and orange indicates suboptimal. ↑: higher is better, ↓: lower is better.

Method	face				girl			
	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓
AdvDM [22]+3DGS	32.63	0.9235	0.2465	0.0710	30.83	0.8716	0.2401	0.0586
Mist [21]+3DGS	33.63	0.9599	0.2539	0.0828	32.63	0.9131	0.2588	0.0872
SDS [36]+3DGS	32.39	0.9707	0.2493	0.0863	34.97	0.9573	0.2631	0.1071
AdvPaint [16]+3DGS	32.95	0.8996	0.2266	0.0355	32.93	0.8514	0.2434	0.0464
DEGauss (Ours)	33.92	0.8860	0.2193	0.0325	35.59	0.8689	0.2380	0.0446
Method	bear				bicycle			
	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓
AdvDM [22]+3DGS	29.26	0.9072	0.3057	0.0378	29.84	0.9385	0.2514	0.0732
Mist [21]+3DGS	31.10	0.9564	0.3106	0.0373	32.52	0.9675	0.2380	0.0567
SDS [36]+3DGS	31.30	0.9546	0.3107	0.0378	32.59	0.9724	0.2415	0.0616
AdvPaint [16]+3DGS	30.77	0.9046	0.3011	0.0314	30.89	0.9257	0.2464	0.0576
DEGauss (ours)	32.20	0.8949	0.2940	0.0256	32.84	0.9358	0.2336	0.0535

clearly illustrate that our DEGauss is able to effectively suppress malicious editing while maintaining a higher degree of visual consistency with the original 3D scene.

4.3 Generalization

To further evaluate the generalization of our DEGauss, we conduct experiments on SOTA different 3D editing methods, including GaussianEditor [7], DGE [4], DreamCatalyst [18] and EditSplat [19]. As shown in Fig. 4, our DEGauss preserves the original 3D appearance against GaussianEditor without introducing visible artifacts. In addition, for DGE and EditSplat, although subtle color modifications are observed, our DEGauss effectively prevents structural editing and maintains overall geometric

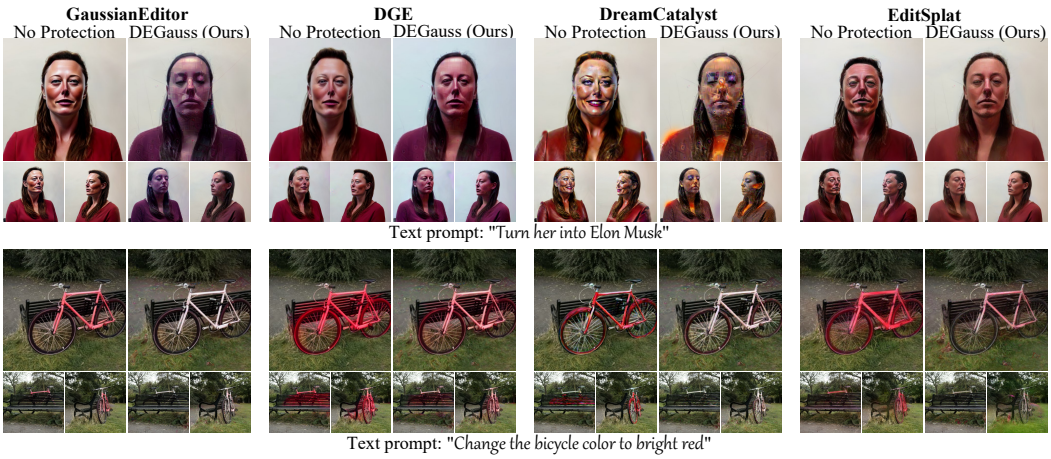

Figure 4: Visualization of generalization experiments. The editing methods include GaussianEditor [7], DGE [4], DreamCatalyst [18], and EditSplat [19]. Our DEGauss significantly resist 3D editing from different existing schemes.

Table 2: Quantitative analysis of generalization experiments. Our DEGauss is outstanding against different editing schemes. "Ori" and "Ours" refer to the non-protected and our protection results, and "Diff" refers to the difference between "Ori" and "Ours". ↑: higher is better, ↓: lower is better.

	face								
	CLIP			CLIP-T			CLIP-D		
Method	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑
GaussianEditor [7]	1.000	0.886	0.114	0.284	0.219	0.065	0.084	0.033	0.051
DGE [4]	1.000	0.890	0.110	0.253	0.230	0.023	0.089	0.050	0.039
DreamCatalyst [18]	1.000	0.896	0.104	0.277	0.248	0.029	0.089	0.044	0.045
EditSplat [19]	1.000	0.933	0.067	0.248	0.236	0.012	0.083	0.053	0.030
	girl								
	CLIP			CLIP-T			CLIP-D		
Method	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑
GaussianEditor [7]	1.000	0.869	0.131	0.260	0.238	0.022	0.120	0.045	0.075
DGE [4]	1.000	0.919	0.081	0.259	0.250	0.009	0.119	0.067	0.052
DreamCatalyst [18]	1.000	0.821	0.179	0.265	0.244	0.021	0.096	0.064	0.032
EditSplat [19]	1.000	0.942	0.058	0.276	0.265	0.011	0.124	0.085	0.039
	bear								
	CLIP			CLIP-T			CLIP-D		
Method	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑
GaussianEditor [7]	1.000	0.895	0.105	0.313	0.294	0.019	0.062	0.026	0.036
DGE [4]	1.000	0.890	0.110	0.310	0.298	0.012	0.071	0.023	0.048
DreamCatalyst [18]	1.000	0.917	0.083	0.333	0.305	0.028	0.074	0.035	0.039
EditSplat [19]	1.000	0.935	0.065	0.312	0.309	0.003	0.051	0.028	0.023
	bicycle								
	CLIP			CLIP-T			CLIP-D		
Method	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑	Ori	Our↓	Diff↑
GaussianEditor [7]	1.000	0.939	0.061	0.270	0.234	0.036	0.076	0.054	0.022
DGE [4]	1.000	0.956	0.044	0.293	0.261	0.032	0.098	0.088	0.010
DreamCatalyst [18]	1.000	0.955	0.045	0.269	0.243	0.026	0.092	0.057	0.035
EditSplat [19]	1.000	0.951	0.049	0.280	0.272	0.008	0.116	0.093	0.023

and semantic integrity. As for DreamCatalyst, our perturbations lead to the generation of noisy and chaotic outputs, significantly disrupting the intended editing direction. These results collectively demonstrate that our DEGauss robustly resists unauthorized 3D editing in the state-of-the-art editing pipelines, highlighting its strong generalization to diverse editing paradigms.

(a) *w/o* $\mathcal{L}_{\text{FD}}$ (b) *w/o* $\mathcal{L}_{\text{GD}}$ (c) *w/o* Wide-view sampling (d) *w/o* View-focal weighting (e) Full components

Figure 5: Visualization of the ablation on key components. The editing model is GaussianEditor and the text prompt is "Make him wear a Venetian mask".

Table 2 reports the CLIP-based similarity scores between our DEGauss results and the original samples under different 3D editing methods. The samples protected by DEGauss consistently achieve significantly lower similarity under all evaluated editors, indicating that the introduced perturbations effectively disrupt the expected editing trajectories regardless of the editing scheme applied.

4.4 Ablation Studies

Ablation on Key Components. We perform an ablation study on the key components of our framework, the visual results are shown Fig. 5. When the feature discrepancy loss is removed, the edited images exhibit noticeable degradation in quality. While without the guidance discrepancy loss, the outputs retain the intended semantic edits, leading to the weakest defense performance. In the absence of wide-view sampling or view-focus weighting, the results inconsistent in multi-view, i.e., some views are effectively defended while others are poorly. In contrast, when all components are integrated, our method achieves the strongest overall protection against malicious editing.

Ablation on Hyperparameters. We verify the effect of noise strength on stealthiness by uniformly adjusting the perturbation hyperparameter λ (jointly λ_{FD} and λ_{GD}), as shown in Table 3. It can be observed that reducing λ improves noise stealthiness (higher PSNR score), while decreasing defense performance (higher CLIP scores). When $\lambda = 1e\text{-}5$, noise invisibility and defense ability reach a trade-off.

Table 3: Noise Stealthiness vs. Defense Strength. Red indicates optimal and orange indicates suboptimal.

	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓
$1e$-4	27.47	0.8147	0.2018	0.0171
$1e$-5	33.92	0.8860	0.2193	0.0325
$1e$-6	34.80	0.9617	0.2692	0.0555

We conduct an ablation study on the hyperparameters of loss function, as shown in Table 4. When the weight of the feature discrepancy loss is reduced, the PSNR score drops to lowest, indicating poor visual quality in the generated samples. Alternatively, decreasing the weight of the guidance discrepancy loss leads to the highest PSNR, but the defense becomes ineffective and the edited result will retain more of the expected semantics. In contrast, our chose hyperparameter setting achieves a balanced trade-off between visual fidelity and editing resistance.

Table 4: Ablation experiments on hyperparameters. The editing model is GaussianEditor. Red indicates optimal and orange indicates suboptimal.

λ_{FD}	λ_{GD}	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓
*0.1	-	31.56	0.8825	0.2297	0.0594
-	*0.1	36.17	0.9244	0.2295	0.1020
-	-	33.92	0.8860	0.2193	0.0325

5 Conclusions

In this paper, we propose DEGauss, a novel defense framework dedicated to protecting 3DGS from malicious editing. Unlike 2D defense approaches that focus only on image space, our DEGauss optimizes perturbations in 3D space to maintain consistent protection across multiple views. Specifically, we elaborate the view-focal gradient fusion mechanism and the dual-discrepancy optimization strategy that jointly disturb the direction of multi-view editing during iterative editing, thus preventing the expected semantic editing. Experimental results indicate that our DEGauss can effectively defend unauthorized 3D editing in different scenes while maintaining high fidelity visual quality.

Acknowledgments

The authors are very indebted to the anonymous referees for their critical comments and suggestions for the improvement of this paper. This work was supported by the National Key Research and Development Program of China (2021YFA1000102), National Natural Science Foundation of China (Nos. 62376285, 61673396), Natural Science Foundation of Shandong Province (No: ZR2022MF260).

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5460–5469, 2022.

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[3] Junkun Chen and Yuxiong Wang. ProEdit: Simple Progression is All You Need for High-Quality 3D Scene Editing. In *Advances in Neural Information Processing Systems*, volume 37, pages 4934–4955, 2024.

[4] Minghao Chen, Iro Laina, and Andrea Vedaldi. DGE: Direct Gaussian 3D Editing by Consistent Multi-view Editing. In *European Conference on Computer Vision*, page 74–92, 2024.

[5] Ruoxi Chen, Haibo Jin, Yixin Liu, Jinyin Chen, Haohan Wang, and Lichao Sun. EditShield: Protecting Unauthorized Image Editing by Instruction-guided Diffusion Models. In *European Conference on Computer Vision*, page 126–142, 2024.

[6] Xinquan Chen, Xitong Gao, Juanjuan Zhao, Kejiang Ye, and Cheng-Zhong Xu. AdvDiffuser: Natural Adversarial Example Synthesis with Diffusion Models. In *IEEE/CVF International Conference on Computer Vision*, pages 4562–4572, 2023.

[7] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.

[8] June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. DiffusionGuard: A Robust Defense Against Malicious Diffusion-based Image Editing. In *International Conference on Learning Representations*, 2025.

[9] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Transactions on Graphics*, 41(4):141:1–141:13, 2022.

[10] Zhongliang Guo, Lei Fang, Jingyu Lin, Yifei Qian, Shuai Zhao, Zeyu Wang, Junhao Dong, Cunjian Chen, Ognjen Arandjelović, and Chun Pong Lau. A Grey-box Attack against Latent Diffusion Model-based Image Editing by Posterior Collapse. *arXiv preprint arXiv.2408.10901*, 2024.

[11] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions. In *International Conference on Computer Vision*, pages 19683–19693, 2023.

[12] Kai He, Chin-Hsuan Wu, and Igor Gilitschenski. CTRL-D: Controllable Dynamic 3D Scene Editing with Personalized 2D Diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 574, pages 6840–6851, 2020.

[14] Robert Hönig, Javier Rando, Nicholas Carlini, and Florian Tramèr. Adversarial Perturbations Cannot Reliably Protect Artists From Generative AI. In *International Conference on Learning Representations*, 2025.

[15] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Liangliang Cao, and Shifeng Chen. Diffusion Model-Based Image Editing: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–27, 2025.

[16] Joonsung Jeon, Woo Jae Kim, Suhyeon Ha, Sooel Son, and Sung-eui Yoon. AdvPaint: Protecting Images from Inpainting Manipulation via Adversarial Attention Disruption. In *International Conference on Learning Representations*, 2025.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[18] Jiwook Kim, Seonho Lee, Jaeyo Shin, Jiho Choi, and Hyunjung Shim. DreamCatalyst: Fast and High-Quality 3D Editing via Controlling Editability and Identity Preservation. In *International Conference on Learning Representations*, 2025.

[19] Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Shin Sangheon, Sangmin kim, and Sangpil Kim. EditSplat: Multi-View Fusion and Attention-Guided Optimization for View-Consistent 3D Scene Editing with 3D Gaussian Splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[20] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. FocalDreamer: Text-Driven 3D Editing via Focal-Fusion Assembly. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3279–3287, 2024.

[21] Chumeng Liang and Xiaoyu Wu. Mist: Towards Improved Adversarial Examples for Diffusion Models. *arXiv preprint arXiv.2305.12683*, 2023.

[22] 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 *International Conference on Machine Learning*, pages 20763–20786, 2023.

[23] Ling Lo, Cheng Yu Yeo, Honghan Shuai, and Wenhuang Cheng. Distraction is All You Need: Memory-Efficient Image Immunization against Diffusion-Based Image Editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24462–24471, 2024.

[24] Guan Luo, Tianxing Xu, Yingtian Liu, Xiaoxiong Fan, Fanglue Zhang, and Songhai Zhang. 3D Gaussian Editing with A Single Image. In *ACM Multimedia*, pages 6627 – 6636, 2024.

[25] Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, and et al. Safety at Scale: A Comprehensive Survey of Large Model Safety. *arXiv preprint arXiv.2502.05206*, 2025.

[26] Tarik Can Ozden, Ozgur Kara, Oguzhan Akcin, Kerem Zaman, Shashank Srivastava, Sandeep P. Chinchali, and James M. Rehg. Optimization-Free Image Immunization Against Diffusion-Based Editing. *arXiv preprint arXiv.2411.17957*, 2024.

[27] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. In *International Conference on Learning Representations*, 2023.

[28] Yansong Qu, Dian Chen, Xinyang Li, Xiaofan Li, Shengchuan Zhang, Liujuan Cao, and Rongrong Ji. Drag Your Gaussian: Effective Drag-Based Editing with Score Distillation for 3D Gaussian Splatting. In *ACM SIGGRAPH*, 2025.

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

[30] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the Cost of Malicious AI-Powered Image Editing. In *International Conference on Machine Learning*, 2023.

[31] Chunyen Shih, Lixuan Peng, Jiawei Liao, Ernie Chu, Chengfu Chou, and Juncheng Chen. Pixel Is Not A Barrier: An Effective Evasion Attack for Pixel-Domain Diffusion Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 6905–6913, 2025.

[32] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. NeRF-Art: Text-Driven Neural Radiance Fields Stylization. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):4983–4996, 2024.

[33] Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-Consistent 3D Editing with Gaussian Splatting. In *European Conference on Computer Vision*, pages 404–420, 2024.

[34] Jing Wu, Jiawang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. GaussCtrl: Multi-View Consistent Text-Driven 3D Gaussian Splatting Editing. In *European Conference on Computer Vision*, page 55–71, 2024.

[35] Tianxing Xu, Wenbo Hu, Yukun Lai, Ying Shan, and Songhai Zhang. Texture-GS: Disentangling the Geometry and Texture for 3D Gaussian Splatting Editing. In *European Conference on Computer Vision*, pages 37–53, Cham, 2024.

[36] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion based mimicry through score distillation. In *International Conference on Learning Representations*, 2024.

[37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[38] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris Metaxas, and Jian Ren. SINE: SINgle Image Editing with Text-to-Image Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6027–6037, 2023.

[39] Yian Zhao, Wanshi Xu, Yang Wu, Weiheng Huang, Zhongqian Sun, and Wei Yang. ProGDF: Progressive Gaussian Differential Field for Controllable and Flexible 3D Editing. *arXiv preprint arXiv.2412.08152*, 2024.

[40] Boyang Zheng, Chumeng Liang, and Xiaoyu Wu. Targeted Attack Improves Protection against Unauthorized Diffusion Customization. In *International Conference on Learning Representations*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, and the claims made are consistent with theoretical and experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a limitations section in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theorems, formulas and proofs in the paper are numbered and cross-referenced, and the theorems and reasoning relied upon have been attributed.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We give a description of all the parameters in the main paper and the algorithm in the Supplementary Material, which will be sufficient to support our algorithmic reproductions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will open access to the code after the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided the experimental setup in Section 4.1 of the paper and a more detailed addition is given in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Instead of including error bars, we selected the average of multiple runs as the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provide sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this thesis is in all respects consistent with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the social impacts in the Supplementary Material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of high-risk models or datasets (e.g., large pretrained language models, generative models, or scraped data). All datasets used are publicly available and licensed for research use.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our assets used in the paper have been noted or cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We did not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.