

Supplementary Material

DEGauss: Defending Against Malicious 3D Editing for Gaussian Splatting

A Additional Preliminary

A.1 3D Gaussian Splatting

3DGS is an explicit novel-view synthesis technique that has gained widespread adoption for 3D scene reconstruction, owing to its efficient rendering capabilities. It models the scene by a set of 3D Gaussian distributions, which can be mathematically represented as:

$$G(x; \mu, \Sigma) = e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}, \quad (1)$$

where $\mu \in \mathbb{R}^3$ is the Gaussian mean used to represent the position and $\Sigma \in \mathbb{R}^{3 \times 3}$ is the covariance matrix used to represent rotation and scaling. During rendering, each Gaussian is projected onto the image plane by rasterization. Given a ray r emitted from the camera, the contribution of a Gaussian to the corresponding pixel is given by:

$$C = \sum_{i \in P} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

where α_i controls transparency, c is color, and the pixels are overlapped by P points in order to calculate the color of the pixel.

A.2 3D Editing with 3DGS

The typical workflow for 3D editing with 3DGS consists of three main phases, as shown in Fig. 1. First, render the multi-view image based on the constructed 3D scene representation. Second, apply editing guidance based on user prompts and edit the multi-view image using a pre-trained diffusion model. Third, the geometric and appearance parameters of the 3D representation are updated to match the editing intent while ensuring multi-view consistency and rendering fidelity across different viewpoints. In addition, the editing is not done overnight, but needs to be done iteratively by re-rendering the image with the edited 3D scene.

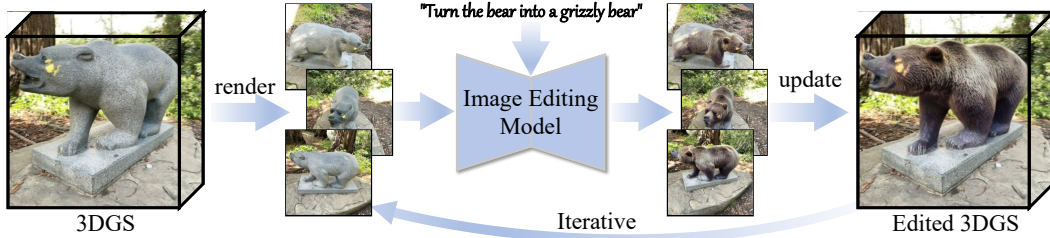


Figure 1: The generalized framework of 3D editing. Rendered images from the established 3DGS are edited by the image editing model, and the edited results are used to iteratively update Gaussian parameters through loss optimization.

B Detailed Experimental Setup

B.1 The Setup of Our DEGauss

In this paper, we employ a pre-trained conditional diffusion-based text-to-image editing model, InstructPix2Pix [1], for generating target view edits extremely featured based on text prompt. The model is implemented using HuggingFace Diffusers¹ official library and works at a resolution of

¹<https://huggingface.co/timbrooks/instruct-pix2pix>

Algorithm 1 The algorithm of our DEGauss.

Require: Initial perturbation Δ , 3D object G with parameter Θ and reference viewpoint set \mathcal{V} , text prompt Y_{lib} , learning rate η , number of sampled views N , number of iterations K

Ensure: Optimized parameter $\Theta^{(k)}$

```

 $\Delta = \{\Delta_\mu, \Delta_\epsilon, \Delta_\alpha, \Delta_c\} \leftarrow \mathbf{0}$  ▷ Initialize perturbation
for  $k = 1$  to  $K$  do
   $V_1 \leftarrow \{v_{\text{ref}} \sim \mathcal{V}\}$  ▷ Randomly sample a reference viewpoint
  for  $n = 2$  to  $N$  do
    for all  $v_j \in \mathcal{V} \setminus V_{n-1}$  do
       $d_j = \min_{v_i \in V_{n-1}} d(v_i, v_j)$  ▷ Compute min distance to current set
    end for
     $v^* = \arg \max_{v_j} d_j$  ▷ Select farthest viewpoint
     $V_n \leftarrow V_{n-1} \cup \{v^*\}$  ▷ Update selected set
  end for
  for all  $v_i \in V_n$  do
     $I_i = \mathcal{R}_{v_i}(\Theta^{(k-1)})$  ▷ Render image
     $\mathcal{L}_{\text{total}}^i = \mathcal{L}_{\text{render}} + \lambda_{\text{FD}} \cdot \mathcal{L}_{\text{FD}} + \lambda_{\text{GD}} \cdot \mathcal{L}_{\text{GD}}$  ▷ Compute overall loss
     $w_i = \frac{(\mathcal{L}_{\text{total}}^i + \epsilon)^\gamma}{\sum_{V_n} (\mathcal{L}_{\text{total}} + \epsilon)^\gamma}$  ▷ Compute focal weights
  end for
   $\Delta^{(k)} \leftarrow \eta \cdot \text{sign}(\sum_{n=1}^N w_n \cdot \nabla_{\Theta^{(k-1)}} \mathcal{L}_{\text{total}}^n)$  ▷ Update perturbation
   $\Theta^{(k)} \leftarrow \Theta^{(k-1)} + \Delta^{(k)}$  ▷ Parameters optimization
end for
return  $\Theta^{(k)}$ 

```

512 × 512. For each scene, we set different prompts for editing based on the target semantics, and the list of prompts is shown in Table 1.

Table 1: List of text prompts for editing each scene.

scenes	Text prompts	scenes	Text prompts
face	Wear him a glasses	girl	Turn her into an old lady
	Make him wear a Venetian mask		Turn her into Elon Musk
	Turn him into the Tolkien Elf		Wear her a glasses
	Turn him into an Einstein		Turn her into the Tolkien Elf
	a photo of a marble sculpture		Turn the woman into a robot
person	Make the man look like a mosaic Sculpture	bear	Turn the bear statue into a panda
	Turn the man into a robot		Turn the bear statue into a grizzly bear
	Make him reading a book		Turn the bear statue into an asiatic black bear
	Turn him into a Minecraft character		Turn the bear statue into a wild boar
bicycle	Change the bicycle color to bright red	garden	Change the table color to deep mahogany brown
	Turn the ground into a Namibian desert		Cover the grass with autumn leaves
	A photo of the bicycle at the namibian desert		A photo of a garden scene with autumn

For each scene, we reconstruct the initial 3D representation G from image sequences using COLMAP, initialize the Gaussian parameters as in previous studies, and train for 30,000 rounds. Optimization is performed using the Adam optimizer with the learning rate to be the same in vanilla 3DGS [5]. Our experiments are running on Ubuntu 22.04.1 with hardware including: CPU Intel(R) Core(TM) i5-13600KF, one GPU NVIDIA RTX 4090 and 32G RAM. We will open access to the code after the paper is accepted. The algorithm is shown in Alg. 1.

B.2 The Setup of Baselines and Editing Methods

To evaluate the applicability of the 2D defense approach in 3D scenes, we adapt the 2D perturbation-based technique and evaluate it in a multi-view setup. Specifically, we converted the original optimizations and updates for 2D perturbation to updates for 3DGS parameters in the same way as we did. In addition, for a fair comparison, we also set the limits of the perturbation to obtain PSNR values similar to those in this paper.

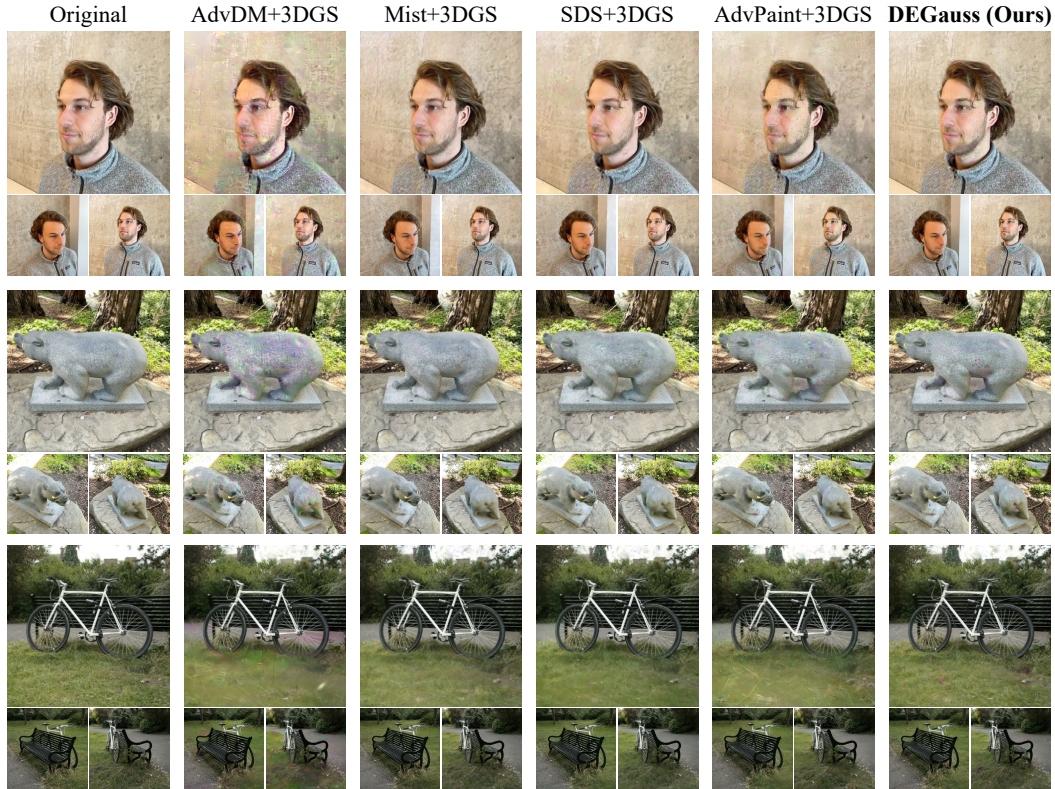


Figure 2: Visual display of protected samples. We show that our DEGauss is visually more natural and less perturbed for protected samples compared to existing methods.

To ensure fair demonstration of generalizability, we implemented and evaluated all 3D editing methods under consistent experimental conditions. Concretely, we follow their original configurations whenever applicable, including the guidance scale and the number of iteration turns, and utilize the same reconstructed 3D scene and editing cues as in our method.

C Additional Experiments

C.1 Visual Display of Protected Samples

We present visual comparisons of protected 3D samples generated by our DEGauss, against representative baselines. As shown in Fig. 2, previous methods usually introduce noticeable artifacts, oversaturation or unnatural distortions, especially in semantically important regions. On the contrary, the protected 3D scene generated by our DEGauss are visually more natural, maintaining the structural integrity, semantic content and realistic appearance of the original scene. This is mainly due to our targeted view-focal fusion strategy, which averages the perturbation constraints in more evenly while ensuring multi-view consistency.

C.2 Additional Comparisons

To further validate the visual effectiveness of our DEGauss, we provide a qualitative comparison with an existing protection baseline under different editing prompts and scenes, as shown in Fig. 3. In a variety of scenarios, including preserve identity (e.g., “wear a glasses”) and change appearance (e.g., “turn into elf”), our DEGauss shows minimal visible distortion while successfully resisting unauthorized modifications. In contrast, samples generated by baseline methods often fail to protect or show significant visual degradation. In addition, we provide supplementary quantitative comparisons in Table 2. As illustrated, our method consistently outperforms existing baselines across multiple metrics, demonstrating superior protection effectiveness while maintaining high visual quality.

Table 2: Additional quantitative comparison with existing methods. The editing method used is GaussianEditor. **Red** indicates optimal and **orange** indicates suboptimal. \uparrow : higher is better, \downarrow : lower is better.

Method	person				garden			
	PSNR \uparrow	CLIP \downarrow	CLIP-T \downarrow	CLIP-D \downarrow	PSNR \uparrow	CLIP \downarrow	CLIP-T \downarrow	CLIP-D \downarrow
AdvDM [9]+3DGS	26.78	0.8787	0.2566	0.0935	33.56	0.9788	0.2529	0.0363
Mist [8]+3DGS	27.42	0.9027	0.2737	0.1276	35.48	0.9846	0.2456	0.0285
SDS [10]+3DGS	27.39	0.9335	0.2551	0.1100	36.71	0.9905	0.2445	0.0271
AdvPaint [4]+3DGS	27.41	0.8669	0.2463	0.0838	32.77	0.9671	0.2488	0.0213
DEGauss (Ours)	27.73	0.8657	0.2453	0.0742	37.54	0.9743	0.2336	0.0207

Table 3: Additional quantitative analysis of generalization experiments. Our DEGauss is remarkable in the face of different editing schemes. \uparrow : higher is better, \downarrow : lower is better.

Method	Volume	person			garden		
		CLIP \downarrow	CLIP-T \downarrow	CLIP-D \downarrow	CLIP \downarrow	CLIP-T \downarrow	CLIP-D \downarrow
GaussianEditor [3]	CVPR'24	0.8657	0.2453	0.0742	0.9743	0.2336	0.0207
DGE [2]	ECCV'24	0.9273	0.2667	0.1365	0.9659	0.2787	0.0818
DreamCatalyst [6]	ICLR'25	0.8669	0.2712	0.1284	0.9711	0.2665	0.0861
EditSplat [7]	CVPR'25	0.9099	0.2848	0.1243	0.9592	0.2972	0.0744

C.3 Additional Generalization

To further assess the generalization capability of our DEGauss, we display additional experiments. As illustrated in Fig. 4 and summarized in Table 3, our approach consistently delivers a high level of preservation performance across different scenes. Notably, it effectively maintains the semantic integrity of human subjects with fine-scale facial and body details, while simultaneously preserving large-scale spatial layouts and complex textures in scenes. These results demonstrate the adaptability of our approach to a wide range of real-world 3D environments.

C.4 Integrate VFGF into Baselines

Our View-Focal Gradient Fusion strategy (VFGF), include wide-view sampling and view-focal weighting, can be integrated into the baseline methods to boost their performance, as shown in Table 4. It can be seen that our strategy improves the attack performance of all baseline methods, validating the effectiveness of this strategy in multi-view attacks. However, these enhanced baseline methods still appear inferior to our DEGauss, which further highlights the comprehensive advantage of our method.

Table 4: Quantitative comparison of baseline methods with and without the integration of our View-Focal Gradient Fusion (VFGF). **Bold** indicates the best result.

		AdvDM	+VFGF	Mist	+VFGF	SDS	+VFGF	AdvPaint	+VFGF	DEGauss(Ours)
face	CLIP \downarrow	0.9235	0.9131	0.9599	0.9359	0.9707	0.9455	0.8996	0.8920	0.8860
	CLIP-T \downarrow	0.2465	0.2399	0.2539	0.2206	0.2493	0.2314	0.2266	0.2205	0.2193
	CLIP-D \downarrow	0.0710	0.0679	0.0828	0.0792	0.0863	0.0853	0.0355	0.0330	0.0325
bear	CLIP \downarrow	0.9072	0.8993	0.9564	0.9227	0.9546	0.9503	0.9046	0.8984	0.8949
	CLIP-T \downarrow	0.3057	0.2988	0.3106	0.3084	0.3107	0.3036	0.3011	0.2996	0.2940
	CLIP-D \downarrow	0.0378	0.0337	0.0373	0.0349	0.0378	0.0317	0.0314	0.0295	0.0256

C.5 Comparison of Wall Time

We have evaluated the wall-clock optimization time for our DEGauss and all 2D defense methods over a complete optimization of 2000 iterations, using a consistent 3D setting for all experiments. As shown in Table 5, the wall-clock time and GPU memory of our method is slightly higher than that of AdvDM, Mist, and SDS, due to the additional computational overhead introduced by the multi-view weighting strategy. However, our cost is shorter than AdvPaint, as we only compute gradients using a subset of feature maps rather than the full attention map used in AdvPaint.

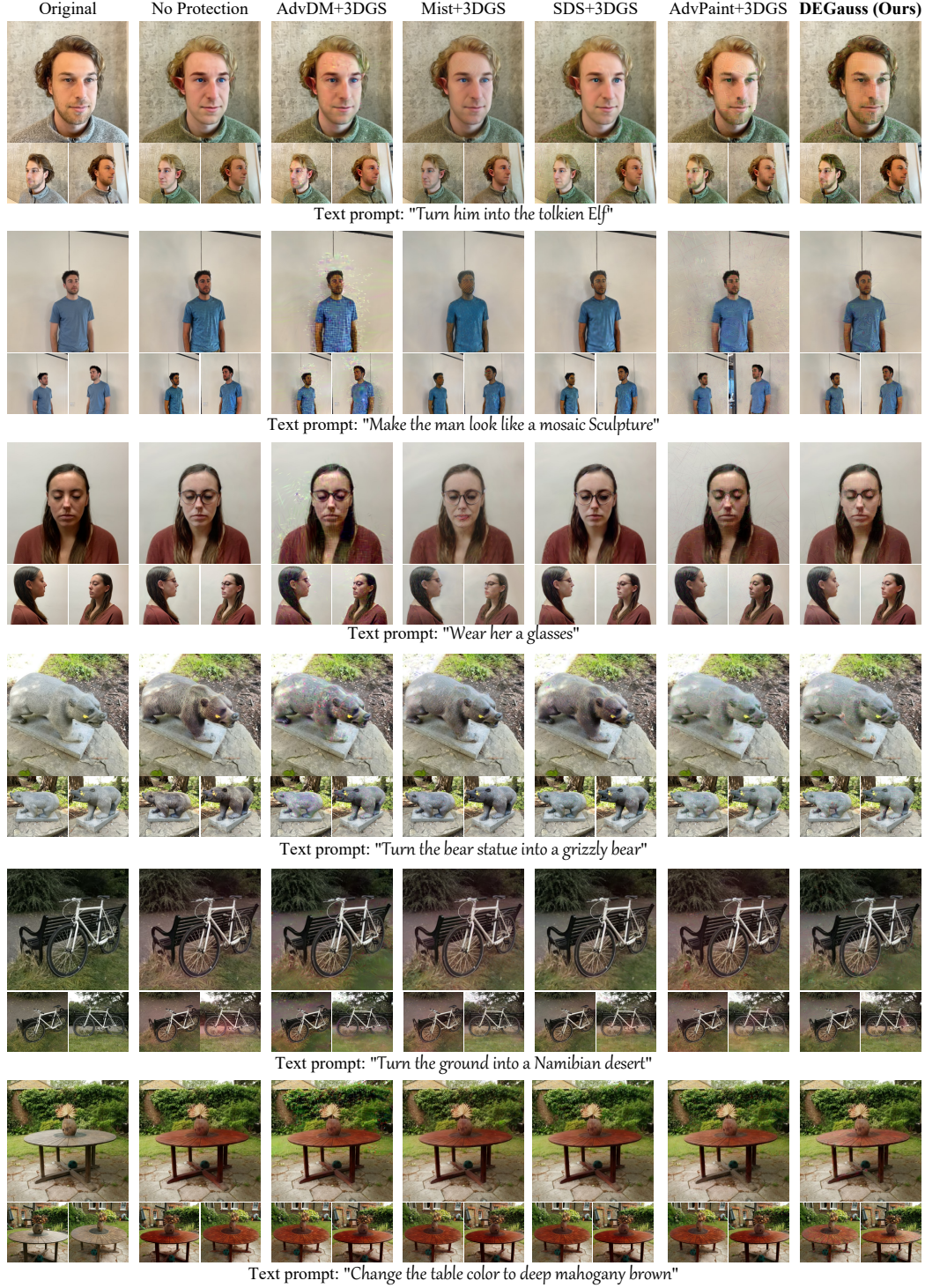


Figure 3: Additional visualizations of comparison experiments.

It is important to note that, although some 2D-based baseline methods may achieve shorter time in simpler settings, they fail to meet the multi-view consistency requirements necessary for effective 3D protection (higher CLIP scores). In contrast, our DEGauss framework achieves superior protection across all views, while maintaining comparable computational cost to other methods.

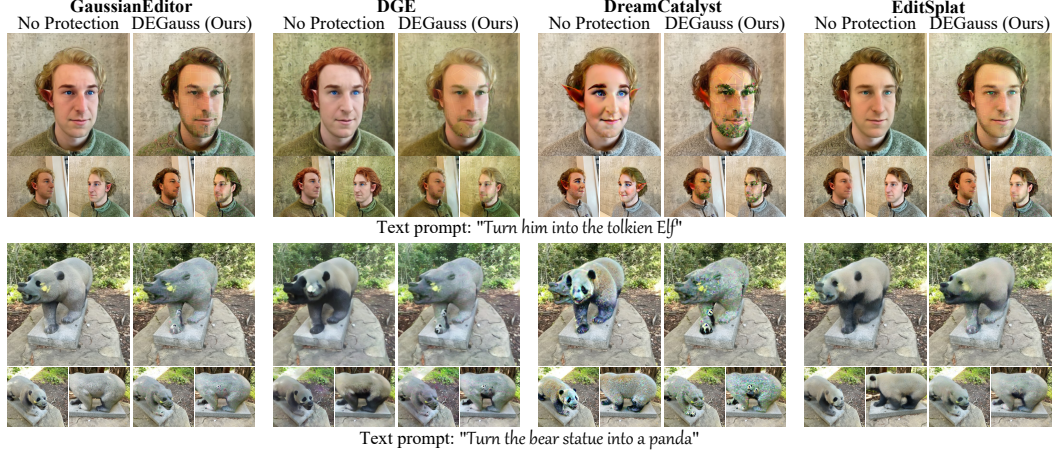


Figure 4: Additional visualizations of generalization experiments.

Table 5: Comparison of wall-clock time and defense performance with 2D defense methods. **Bold** indicates the best result.

	AdvDM+3DGS	Mist+3DGS	SDS+3DGS	AdvPaint+3DGS	DEGauss(Ours)
Wall-Clock Time↓	520.06s	675.49s	361.01s	959.08s	893.07s
GPU Memory↓	15112MiB	16744MiB	13508MiB	20220MiB	16938MiB
CLIP↓	0.9235	0.9599	0.9707	0.8996	0.8860

C.6 Ablation Study

Quantitative Ablation Study on Key Components. We further conduct quantitative ablation studies to evaluate the contribution of each key module in our framework. As shown in Table 6, the complete model achieves the best overall results, while removing individual modules leads to significant performance degradation, especially in terms of protection success and visual fidelity. These results highlight the necessity of each module and demonstrate that our complete design achieves a good balance between effective preservation and natural appearance.

Table 6: Quantitative ablation on key components. The editing model is GasussianEditor. **Red** indicates optimal and **orange** indicates suboptimal. ↑: higher is better, ↓: lower is better.

No.	Ablation Method	PSNR↑	CLIP↓	CLIP-T↓	CLIP-D↓
(a)	w/o \mathcal{L}_{FD}	31.88	0.8714	0.2391	0.0761
(b)	w/o \mathcal{L}_{GD}	36.57	0.9286	0.2642	0.2275
(c)	w/o Wide-view sampling	33.40	0.8896	0.2209	0.0337
(d)	w/o View-focal weighting	33.11	0.8945	0.2304	0.0553
(e)	Full components	33.92	0.8860	0.2193	0.0325

Ablation on the Number of Sampled Views. We also investigated the effect of the number of sampled views on the protection performance. As shown in Table 7, too small a number of views (e.g., 1 or 4) leads to insufficient view coverage, which reduces the ability to resist edits and increases multi-view inconsistency. Conversely, while increasing the number of views (e.g., 8) slightly improves robustness, it can also lead to overfitting local artifacts in less important regions, reducing visual quality. It is worth noting that the use of 6 sampled views achieves the best overall balance, providing strong conservation performance while maintaining high visual fidelity and computational efficiency.

D Limitations

While our DEGauss has shown strong effectiveness in defending against malicious 3D editing in a variety of pipelines and scenes, there are still several limitations. First, the defense is currently designed for 3D Gaussian Splatting-based representations, and its applicability to mesh- or NeRF-based 3D assets requires further research. Second, while our perturbations are visually imperceptible

Table 7: Ablation on the number of sampled views. The editing model is GasussianEditor. Red indicates optimal and orange indicates suboptimal. \uparrow : higher is better, \downarrow : lower is better.

Number of Views N	PSNR \uparrow	CLIP \downarrow	CLIP-T \downarrow	CLIP-D \downarrow
1	32.08	0.8688	0.2309	0.0519
4	30.77	0.8675	0.2372	0.0541
6	33.92	0.8860	0.2193	0.0325
8	31.20	0.8762	0.2247	0.0233

in most cases, their potential impact on other downstream 3D tasks (e.g., reconstruction or recognition) remains an open question. We leave these aspects for future work.

E Social Impact

This research addresses the growing need to secure 3D digital content by proposing a defense framework that prevents unauthorized or malicious editing of 3D scenes. As generative 3D editing tools become more powerful and easy to use, our approach can serve as a foundational component for protecting digital assets such as 3D personal identities. In addition, if not properly applied, adversarial perturbations in the scene may trigger downstream tasks to go wrong. Ensuring the responsible deployment and transparent use of such protection techniques is critical to preventing abuse.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Joonsung Jeon, Woo Jae Kim, Suhyeon Ha, Soeul Son, and Sung-eui Yoon. AdvPaint: Protecting Images from Inpainting Manipulation via Adversarial Attention Disruption. In *International Conference on Learning Representations*, 2025.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Chumeng Liang and Xiaoyu Wu. Mist: Towards Improved Adversarial Examples for Diffusion Models. *arXiv preprint arXiv.2305.12683*, 2023.
- [9] 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.
- [10] 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.