Geometry Cloak: Preventing TGS-based 3D Reconstruction from Copyrighted Images

Qi Song[1,2], **Ziyuan Luo**[1,2], **Ka Chun Cheung**[2],
Simon See[2], **Renjie Wan**[1*]

[1]Department of Computer Science, Hong Kong Baptist University
[2]NVIDIA AI Technology Center, NVIDIA
{qisong,ziyuanluo}@life.hkbu.edu.hk
{chcheung,ssee}@nvidia.com, renjiewan@hkbu.edu.hk

Abstract

Single-view 3D reconstruction methods like Triplane Gaussian Splatting (TGS) have enabled high-quality 3D model generation from just a single image input within seconds. However, this capability raises concerns about potential misuse, where malicious users could exploit TGS to create unauthorized 3D models from copyrighted images. To prevent such infringement, we propose a novel image protection approach that embeds invisible geometry perturbations, termed "geometry cloaks", into images before supplying them to TGS. These carefully crafted perturbations encode a customized message that is revealed when TGS attempts 3D reconstructions of the cloaked image. Unlike conventional adversarial attacks that simply degrade output quality, our method forces TGS to fail the 3D reconstruction in a specific way - by generating an identifiable customized pattern that acts as a watermark. This watermark allows copyright holders to assert ownership over any attempted 3D reconstructions made from their protected images. Extensive experiments have verified the effectiveness of our geometry cloak. Our project is available at https://qsong2001.github.io/geometry_cloak.

1 Introduction

With the increasing importance of 3D assets, several methods have been proposed to reconstruct or generate 3D models from single 2D images. Combining with Tensorial Radiance Fields [2], Triplane-based Gaussian Splatting (TGS) [52] presents a compelling approach for producing 3D models from single-view images. However, malicious users could potentially exploit TGS [52] to generate 3D models from single-view images without authorization, posing a threat to the interests of image copyright owners. To address this issue, it is essential for image owners to implement measures that can safeguard their copyrighted images from being used by TGS [52].

Digital watermarking [6, 51] is an effective way of claiming the copyright of digital assets. Thus, one potential method for safeguarding copyrighted images is to embed unique messages within images intended for building a 3D model and then extract the embedded messages from the reconstructed 3D model. However, previous methods have proven it is difficult to transfer the embedded copyright messages in 2D images into 3D models [31, 19]. Moreover, even if we can embed and extract copyright messages, the 3D models might have already been used by others before the copyright being claimed.

To prevent unauthorized 3D reconstruction from copyrighted images via TGS [52], a possible approach is to incorporate adversarial perturbations [11, 28] into input images intended for TGS.

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3793

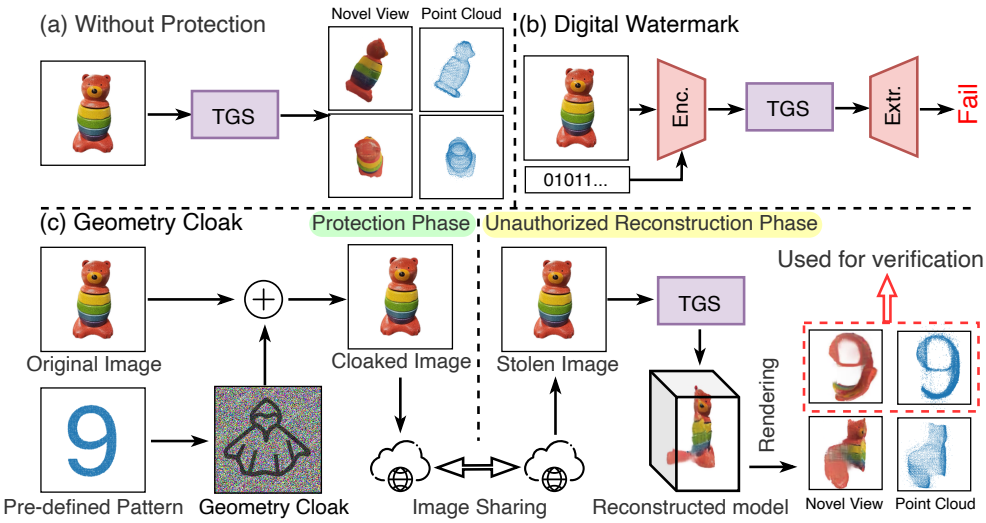

Figure 1: Overview of our scenario. (a) Images without protection. Images can be easily reconstructed into 3D models by malicious users with TGS [52], posing a threat to the copyright of the image owner. (b) Digital Watermarking offers a solution by embedding copyright messages into the view-image before 3D reconstruction. However, the embedded message cannot be extracted from novel rendered views. (c) Geometry Cloak. Our geometry cloak utilizes the disturbance-prone components of TGS, achieving view-specific watermark embedding. Our method can compromise the unauthorized reconstructed 3D model while providing a verifiable pattern for copyright claim.

Adversarial methods [11, 28] have already achieved promising results by introducing disturbances into input images to prevent neural models from functioning correctly. When it comes to TGS [52], a straightforward solution is to incorporate such adversarial perturbations into input images by maximizing the difference between rendered and ground truth views like previous methods [9, 13]. However, those methods [9, 13] intuitively focus on disturbing rendered results, ignoring perturbation-prone components of TGS [52]. As a result, those conventional adversarial perturbations can only lead to limited changes to the reconstructed results [9, 13], which may still be used for several illicit applications. Besides, simple adversarial perturbations can only disrupt the results of 3D reconstruction in an uncontrollable manner but do not support traceability. The users have difficulties claiming copyright after the disturbance-affected 3D generation of images.

We envision a novel scenario where copyrighted images can induce TGS to generate compromised content with an identifiable pattern. To achieve this goal, a naive solution is to employ image cloaking. Traditionally, image cloaking techniques [23, 37, 41, 35] are used to prevent the image from malicious editing by diffusion models [33]. The cloaking integrated into images can shift image features into another domain through adversarial perturbations. Consequently, this manipulation directs diffusion models to generate predetermined specific outcomes. If malicious users attempt to create 3D models using these protected images, the resulting compromised 3D models will be unusable. Besides, the identifiable pattern exhibited can help the image owner assert their copyrights in the event of legal inquiries.

However, unlike image cloak methods [35, 23] against diffusion models [33], simply perturbing image features cannot effectively induce TGS (as shown in Fig. 1). Image features in TGS show strong robustness against disturbances, even under relatively strong attack settings. Thus, the key becomes how to identify perturbation-prone components in TGS [52] and capitalize on this weakness to induce reconstructed results. TGS [52] contains image and geometry features during its 3D model reconstructions. Previous works [8, 3] have shown the geometry feature in 3D Gaussian Splatting [18] is easier to be manipulated with external operations. Therefore, we wonder if it is possible to manipulate the estimated point cloud process with an invisible disturbance. Based on this simple observation, we propose embedding invisible adversarial perturbations as a geometry cloak on images intended for TGS [52], which can affect and manipulate estimated point clouds in the process of TGS's 3D reconstruction.

As shown in Fig. 2, we introduce a geometry cloak, which is carefully crafted to induce TGS [52] to fail 3D model generation and reveal our embedded pattern. To induce the TGS to reveal the embedded patterns, we propose a view-specific Projected Gradient Descent [28] (view-specific PGD) strategy by optimizing the distance between the projected point cloud view and predefined patterns. The PGD iteratively updates the geometry cloak by minimizing CD loss, ultimately revealing the desired view and uncovering the hidden patterns within the TGS [52]. When malicious users reconstruct 3D models using the protected images with TGS, the compromised geometry information causes TGS to reveal the embedded message. Unlike traditional copyright protection methods like digital watermarking [25, 51, 39], which typically entail additional procedures for extracting the watermark post-use. Our approach directly transforms TGS [52] into a message disclosure tool by inducing it to yield a specific stylistic outcome, facilitating the verification of image ownership rights. In summary, our main contributions are threefold:

- We propose the concept of "geometry cloaking", which can prevent unauthorized image-to-3D generation by TGS [52], and it also leaves a verifiable copyright pattern.
- Our geometry cloaking technique explores the perturbation-prone components of Triplane-based Gaussian Splatting and utilizes this vulnerability to achieve the protection of images.
- We propose a view-specific PGD strategy, which can embed identifiable patterns into a specific view of the reconstructed 3D model.

As our approach attacks the geometry features that are widely used in various single-image-to-3D approaches, it demonstrates generalization capability to other GS-based single-view to 3D approaches like LGM [40]. The results can be found in our experiments.

2 Related work

2.1 Building 3D models from single images

There has been a surge of research on learning to generate novel views from a single image [47, 36, 52]. This task aims to infer the 3D structure of a scene from a single 2D image and render new perspectives, enabling applications in virtual reality, augmented reality, and computer-aided design. One approach is NeuralLift-360 [47], which incorporates a CLIP loss [32] to enforce similarity between the rendered image and the input image. Another method, 3DFuse [36], fine-tunes the Stable Diffusion model [33] with LoRA [14] layers and a sparse depth injector. Recent work like Zero123 [24] takes a different approach by fine-tuning the latent stable diffusion model [33] to generate novel views based on relative camera pose. Recently, Triplane-based neural rendering methods [2, 52, 12] adopt a novel approach to model and reconstruct radiance fields. Unlike NeRF, which uses pure MLPs, Tensorial Radiance Fields (TensoRF) [2] consider the full volume field as a 4D tensor and propose to factorize the tensor into multiple compact low-rank tensor components for efficient scene modeling. Combing TensoRF [2] with novel 3D Gaussian Splatting [18], Zou *et al.*propose a Triplane-base Gaussian Splatting [52], which can obtain a 3D model from single-view image within seconds [52]. With continued research and development, more impressive and realistic 3D reconstructions in the future will enable immersive experiences and streamlined design workflows. The advancements in creating 3D models from a single-view image offer significant potential for diverse applications and digital assets. Consequently, it is essential to address the protection of copyrighted images to prevent their misuse in generating 3D models.

2.2 Adversarial attacks for neural rendering

Adversarial attacks [11, 28] have become a significant concern in the field of computer vision [22, 16]. These attacks aim to deceive machine learning models by introducing adversarial perturbations to the input data, leading to incorrect predictions such as misclassification [11]. While initially studied in the context of image classification models, adversarial attacks have also been explored in other domains [35, 23]. For Neural Radiance Fields (NeRFs) [29], several works have proposed methods to perturb or enhance the original NeRF framework. NeRFs are a class of models that can synthesize high-quality 3D scenes from 2D images by learning the scene's volumetric representation and appearance. However, NeRFs are also vulnerable to adversarial attacks. Several recent works have investigated different techniques to perturb or enhance the original NeRF framework using adversarial

 https://doi.org/10.52202/079017-3793

attacks. NeRFool [9] presents an approach to manipulate NeRFs by perturbing the scene's geometry and appearance using adversarial attacks. NeRFTargeted [13] focuses on targeted perturbations in NeRFs, allowing for the optimization of scene parameters to generate desired target images. ShieldingNeRF [46] introduces a technique to protect sensitive information in NeRF-generated views by introducing obfuscating perturbations. These works demonstrate the potential of perturbing NeRFs for various objectives, including adversarial attacks, targeted image manipulation, and privacy protection, contributing to the advancement of NeRF-based models [2, 29, 27] in computer graphics and computer vision tasks. However, with the promising developments in 3DGS [18], research on defending against attacks on Gaussian Splatting renderings remains an area that requires further investigation.

2.3 Image cloaking

With the rapid advancement of AI models, the risk of misuse has raised concerns, particularly in the malicious use of public data [43, 42, 50, 17]. Researchers advocate a proactive approach to prevent such misuse by adding subtle noise to images prior to publication. This technique aims to disrupt attempts at exploitation [21, 26, 20]. An important application of image cloaking is to prevent privacy violations from face recognition systems [38, 4, 15]. By introducing noise patterns to facial regions, these methods degrade face recognition model performance while maintaining visual quality. Image cloaking can also thwart image manipulation through GAN-based techniques like DeepFakes [30] by corrupting latent representations. Recent image cloaking methods [37, 23, 41] focus on protecting the copyrighted image from misuse by stable diffusion models. These methods disrupt artistic mimicry and harmful personalized images, aiming to prevent unauthorized exploitation. For example, GLAZE [37] and AdvDM [23] primarily focus on disrupting artistic mimicry and harmful personalized images generated by text-to-image models[10]. Anti-Dreambooth [41] concentrates on fine-tuning DreamBooth [34] for malicious face editing. These techniques aim to cloak input images in a way that disrupts the model's ability to generate personalized content while preserving the overall visual quality. Existing methods focus on protecting the copyrighted image from misuse by disturbing image features. By contrast, our method focuses on preventing copyrighted from being 3D reconstructed by TGS [52] without authorization, facing 3D scenes and complex copyright validity verification settings.

3 Preliminaries of TGS

TGS [52] introduces a novel hybrid 3D representation that integrates an explicit point cloud with an implicit triplane, enabling efficient and high-quality 3D object reconstruction from single-view images. As shown in Fig. 2, the representation consists of a point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ providing explicit geometry, and a triplane $T \in \mathbb{R}^{3\times C\times H\times W}$ encoding an implicit feature field, where $T = (T_{xy}, T_{xz}, T_{yz})$ comprises three orthogonal feature planes. For a given position $x \in \mathbb{R}^3$ from the point cloud, the corresponding triplane feature f_t is obtained by trilinear interpolation and concatenation of features from the three planes:

$$f_t = \text{interp}(T_{xy}, \mathcal{P}_{xy}) \oplus \text{interp}(T_{xz}, \mathcal{P}_{xz}) \oplus \text{interp}(T_{yz}, \mathcal{P}_{yz}). \tag{1}$$

Utilizing this hybrid representation, the 3D Gaussian attributes like opacity α, anisotropic covariance (scale s and rotation q), and spherical harmonics coefficients sh are decoded from f_t augmented with projected local image features f_l using an MLP ϕ_g:

$$(\Delta x', \alpha, s, q, sh) = \phi_g(x, f_t \oplus f_l). \tag{2}$$

Together, these parameters parameterize the 3D Gaussian kernel attributes around the point x, enabling differentiable Gaussian splatting for rendering.

TGS represents a cutting-edge approach to 3D object reconstruction from single-view images, combining explicit and implicit representations to achieve accurate and detailed reconstructions. By leveraging advanced decoding mechanisms and efficient rendering techniques, the model demonstrates superior performance in generating realistic 3D models with intricate geometry and textural details.

4 Methodology

Overview. With the increasing capabilities of 3D reconstruction techniques like TGS [52], there is a risk of malicious users exploiting these methods to generate 3D models from copyrighted images

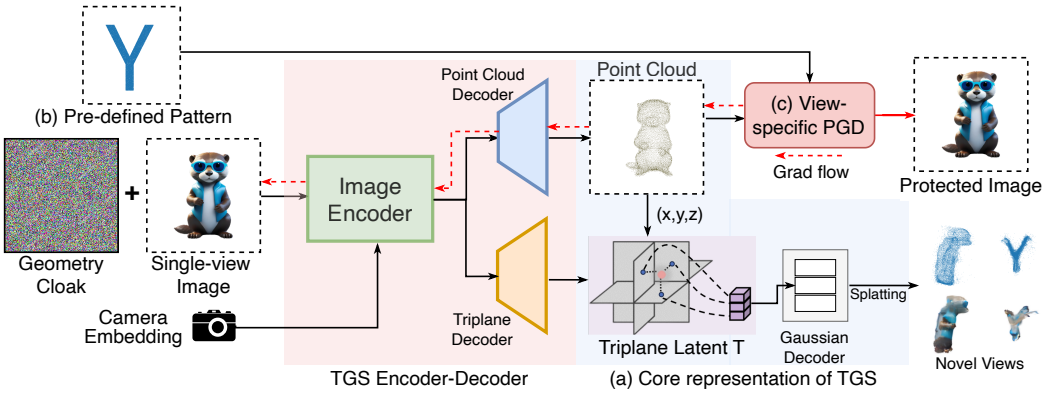

Figure 2: Overall of our proposed method. We propose to induce the 3D reconstruction process with our geometry cloak. (a) The core representation of TGS [52] includes an explicit point cloud and an implicit triplane-based feature field. The features of the novel view image are extracted through the coordinates in the point cloud. (b) The target patterns (Section 4.1) are designed to induce the final reconstruction result. (c) In order to make the reconstruction result show some distinguishable characteristics, we use projected gradient descent (PGD) [28] to iteratively optimize the reconstructed point cloud so that it has consistent characteristics with the target point cloud (Section 4.2).

without authorization, infringing on the rights of image owners. We propose geometry cloak, a novel solution by embedding invisible perturbations on the input images intended for TGS [52]. These perturbations are crafted to induce TGS to fail the reconstruction in a distinct way, producing an identifiable pattern in the corrupted 3D output.

This section presents the methodology of inhibiting 3D reconstruction of TGS [52] via the introduced geometry cloak. Our method consists of two stages: (1) Building verifiable geometry patterns (Section 4.1), and (2) Optimizing geometry cloak with view-specific PGD (Section 4.2).

4.1 Building verifiable geometry pattern

As illustrated in Fig. 2, to obtain a 3D model from single-view images, TGS [52] encodes the single-view image $\mathcal{I}$ and its associated camera parameters into image features. Following this, a point cloud decoder is adopted to project image features onto the point cloud. Then, a triplane decoder converts the image feature into the triplane latent T. Finally, 3D Gaussians are decoded from triplane feature f_t for novel view synthesis.

Our methodology diverges by targeting the explicit geometry features of the point cloud. Perturbing the point cloud directly is inherently more effective due to the inherent vulnerabilities in the TGS reconstruction process. In TGS [52], the point cloud offers a distilled and direct representation [18] of the scene's structure, making modifications on them quite evident [8, 3]. Besides, the point cloud is not only foundational geometry information but also typically helps subsequent processing to obtain the final output. Point clouds provide essential geometry information that is instrumental in the sampling of features from the latent triplane representation (Eq. (1)). Moreover, by introducing strategic alterations to the geometry feature, we bypass some of the inherent robustness found in image-level features (Table 1) and avoid the complex transformations between image and geometry spaces. This direct manipulation allows for more precise control over the adversarial impact, exploiting specific vulnerabilities in TGS and leading to more pronounced disruptions in its output.

Different types of target geometry pattern. Based on the inherent vulnerabilities of point clouds, we propose two types of verifiable pattern (Fig. 3), including (1) *Pre-defined patterns* and (2) *Customized patterns*. Through these patterns, the reconstructed 3D model undergoes consistent changes in geometry and visual appearance (novel views).

Pre-defined patterns are 2D point clouds that are transformed from alphanumeric characters images, creating a direct and straightforward representation of watermarks. To obtain these 2D point clouds, we segment the image of alphanumeric characters and record the coordinates of these alphanumeric

https://doi.org/10.52202/079017-3793

Figure 3: Two different target geometry patterns. (1) Pre-defined patterns: we directly convert alphanumeric characters into a 2D point cloud as watermarks. (2) Customized patterns: In E_1, we first extract the point cloud of the image that needs to be protected. In E_2, we edit the acquired point cloud through text-guided methods like instructP2P [48] or open-source software meshlab [5].

characters' segmentation. Then, we sample the point from segmentation coordinates and form a 2D point cloud as pre-defined patterns.

Customized patterns are a more personalized approach where users can selectively protect a certain part of an image. In this setting, we first extract a point cloud from the image that requires safeguarding. We adopt the same methods from TGS [52] to obtain the point cloud $\hat{\mathcal{P}}$ in E_1. In E_2, users can refine this point cloud to select the area they wish to protect. Users can employ text-guided editing techniques like instructP2P [48] or turn to open-source software like MeshLab [5] to further shape and customize the point cloud. This process not only enhances the visual appeal of the point cloud but also embeds a layer of security by aligning it with specific textual instructions or user intents.

4.2 Optimizing geometry cloak with view-specific PGD

In this section, by exploiting the vulnerability of TGS [52], we present view-specific PGD to determine geometry cloak δ, which can minimize the distance between reconstructed point clouds in TGS and target pre-defined patterns.

Geometry Cloak δ. Geometry cloak δ is crafted to mislead the TGS into a controlled 3D reconstruction with imperceptible perturbations into single-view images. It minimizes the difference to the target geometry pattern while preserving image fidelity perceptible to the human eye through adversarial training [28].

Specifically, our geometry cloak δ aims to minimize the Chamfer Distance $\mathcal{L}_{\text{CD}}$ between point cloud $\hat{P}$ from image $\mathcal{I}$ via TGS and target geometry pattern $\mathcal{P}_{\text{tar}}$ in a certain view. The overall objective can be expressed as follows:

$$\delta := \arg \min_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}_{\text{CD}}(\mathcal{E}(\mathcal{I}+\delta), \mathcal{P}_{\text{tar}}), \tag{3}$$

where $\mathcal{E}$ denotes the network that maps image into point cloud in TGS [52], and ϵ represents the perturbation budget.

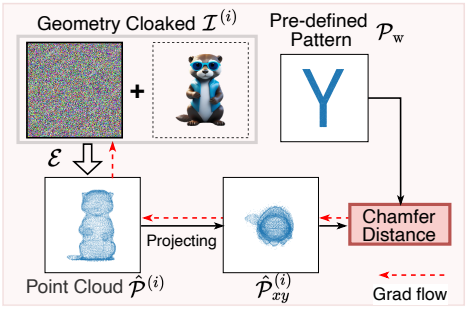

Figure 4: Example of View-specific PGD. We use a 2D point cloud pre-defined pattern $\mathcal{P}_{\text{w}}$ as the target geometry pattern. The watermark is embedded at the viewing direction $\theta = xy$.

View-specific PGD. To embed pre-defined patterns in a specific viewing direction, we develop a view-specific PGD. The optimization iteratively adjusts the geometry cloak δ to make the projected point cloud closer to the pre-defined patterns while keeping the image visually similar to the source. The updates use gradient decent [28] on $\mathcal{L}_{\text{CD}}$ with learning rate α, moving the cloaked image towards the target. This iterative process manipulates the TGS's perception while maintaining visual similarity as below:

$$\mathcal{I}^{(i+1)} = \mathcal{I}^{(i)} + \alpha \cdot \text{sgn}(\nabla_{\mathcal{I}} \mathcal{L}_{\text{CD}}(\hat{\mathcal{P}}^i_\theta, \mathcal{P}_{\text{w}})), \tag{4}$$

where $\hat{\mathcal{P}}^i_\theta = \texttt{Proj}_\theta(\mathcal{E}(\mathcal{I}^{(i)} + \delta^{(i)})$, denoting the projected point cloud at viewing direction θ from cloaked image $(\mathcal{I}^{(i)} + \delta^{(i)})$, and $\mathcal{P}_{\text{w}}$ stands a 2D point cloud watermarks. We encapsulate the geometry cloaking process in Algorithm 1.

https://doi.org/10.52202/079017-3793

Algorithm 1: Optimizing Geometry Cloak with view-specific PGD

Input: Input image $\mathcal{I}$, Point cloud encoder $\mathcal{E}$ from TGS, pre-defined pattern $\mathcal{P}_{\text{w}}$
number of steps N, step size α, perturbation budget ϵ, viewing direction θ
Output: Geometry cloaked image $\hat{\mathcal{I}}$
Initialize geometry cloak $\delta \leftarrow 0$, and geometry cloaked image $\hat{\mathcal{I}} \leftarrow \mathcal{I}$
for $i \leftarrow 1$ **to** N **do**
  $\hat{\mathcal{P}} \leftarrow \mathcal{E}(\hat{\mathcal{I}})$ `// Estimate point cloud representations;`
  $\hat{\mathcal{P}}_\theta \leftarrow \texttt{Proj}_\theta(\hat{\mathcal{P}})$ `// Project the point cloud at view direction` θ;
  $Loss \leftarrow \mathcal{L}_{\text{CD}}(\hat{\mathcal{P}}_\theta, \mathcal{P}_{\text{w}})$ `// Calculate CD between two 2D point clouds;`
  $\delta \leftarrow \alpha \cdot \text{sgn}(\nabla_{\hat{\mathcal{I}}} Loss)$ `// Update geometry cloak;`
  $\hat{\mathcal{I}} \leftarrow \hat{\mathcal{I}} + \delta$ `// Update cloaked image;`
  $\hat{\mathcal{I}} \leftarrow \texttt{clip}(\hat{\mathcal{I}}, \mathcal{I} - \epsilon, \mathcal{I} + \epsilon)$ `// Clip cloaked image` $\hat{\mathcal{I}}$;
end
Return $\hat{\mathcal{I}}$

4.3 Implementation details

Our method uses the PyTorch framework on a single NVIDIA V100 GPU. Our geometry cloak is obtained through optimization (Section 4.2) using projected gradient descent [11]. We adopt a mask version of PGD [28] for calculating geometry cloak, as only the object in the image is used for 3D reconstruction without its background. The input image $\mathcal{I}^{(0)} = \mathcal{I}$ is initialized, and iteratively updated for $N = 100$ steps with a step size of $\alpha = 0.001$. The loss is defined as the Chamfer Distance [1] between the predicted point cloud $\mathcal{P}$ and the target point cloud $\hat{\mathcal{P}}_{tar}$. For pre-defined patterns, we adopt the proposed view-specific PGD as we want to embed the 2D pre-defined patterns into a certain viewing direction θ. For customized patterns, we directly calculate the distance between customized patterns and predicted point clouds, as both are 3D point clouds. The updated image $\mathcal{I}^{(i)}$ is clipped to the valid range $[0, 1]$, and after $N = 100$ iterations, the final image $\hat{\mathcal{I}} = \mathcal{I}^{(N)}$ are obtained as the geometry cloaked output.

5 Experiments

5.1 Settings

Dataset. To evaluate the performance of our method, we conduct experiments on the Google Scanned Objects (GSO) [7] and OmniObject3D (Omni3D) [45] datasets. Both datasets embody a large diversity of view images and provide a rich and varied set of data for assessing the performance of our method. For GSO [7], we select 1 image view for each of the 1030 objects, resulting in 1030 images. For Omni3D [45], we choose 5 image views from each of the 190 classes in Omni3D, resulting in a total of 950 images.

Baselines. To investigate the effectiveness of our approach, we evaluate the impact of different perturbations on the reconstructed results by TGS [52]. We experiment with four types of perturbations strategy: (1) **Gauss. noise**, *i.e.*, random Gaussian noise is added to the protected image. (2) **Adv. image**, *i.e.*, adversarial attacks on image feature [9]; (3) **Geometry cloak w/o target**, *i.e.*, adversarial attacks on the point cloud. For Adv. image and geometry cloak w/o target, we directly use the norm value of features [35] as the optimizing loss. (4) **Geometry cloak**. For the geometry cloak, we randomly select letters "A-Z" and numbers "1-9" as the pre-defined pattern.

Evaluation methodology. We report the quantitative metric between no perturbation and perturbed reconstructed results. Specifically, for visual quality, we report image similarity metrics: PSNR, SSIM [44], LPIPS [49]. For geometry quality, we report the Chamfer Distance (CD) [1]. We further present the qualitative customized 3D reconstruction by inducing the point from the single-view image into another domain.

https://doi.org/10.52202/079017-3793

Table 1: Quantitative comparison of perturbation strategies We present the outcomes of four distinct perturbation strategies compared to the non-perturbed results. These strategies include Gaussian noise (random Gaussian noise), Adversarial image (perturbing image features), geometry cloak without target, and geometry cloak. We evaluate the image quality metrics (PSNR/SSIM/LPIPS) and geometry quality metric (Chamfer Distance, CD) on the Omni3D [45] and GSO [7] datasets at perturbation budgets of $\epsilon = 2, 4, 8$.

ϵ	Perturb. strategy	Omni3D				GSO			
		PSNR ↓	SSIM ↓	LPIPS ↑	CD ↑	PSNR ↓	SSIM↓	LPIPS ↑	CD ↑
2	Gauss. noise	23.35	0.8954	0.1141	4.913	24.24	0.9049	0.1130	5.070
	Adv. image	19.32	0.8424	0.1782	13.86	19.94	0.8478	0.1821	21.89
	Ours w/o target	11.69	0.7513	0.2585	237.7	13.07	0.7849	0.2499	267.5
	Ours	12.47	0.7392	0.2669	337.0	14.06	0.7673	0.2578	352.6
4	Gauss. noise	22.11	0.8788	0.1313	6.424	22.66	0.8847	0.1354	7.940
	Adv. image	19.21	0.8417	0.1784	14.84	19.81	0.8466	0.1827	13.60
	Ours w/o target	11.49	0.7513	0.2603	260.1	12.94	0.7848	0.2504	284.8
	Ours	12.45	0.7400	0.2665	344.9	14.04	0.7669	0.2585	356.5
8	Gauss. noise	20.34	0.8443	0.1651	10.20	20.68	0.8479	0.1724	13.60
	Adv. image	19.17	0.8412	0.1786	13.86	19.80	0.8468	0.1824	23.58
	Ours w/o target	11.44	0.7515	0.2608	273.3	12.89	0.7840	0.2508	296.3
	Ours	12.44	0.7396	0.2664	348.4	14.02	0.7661	0.2584	358.7

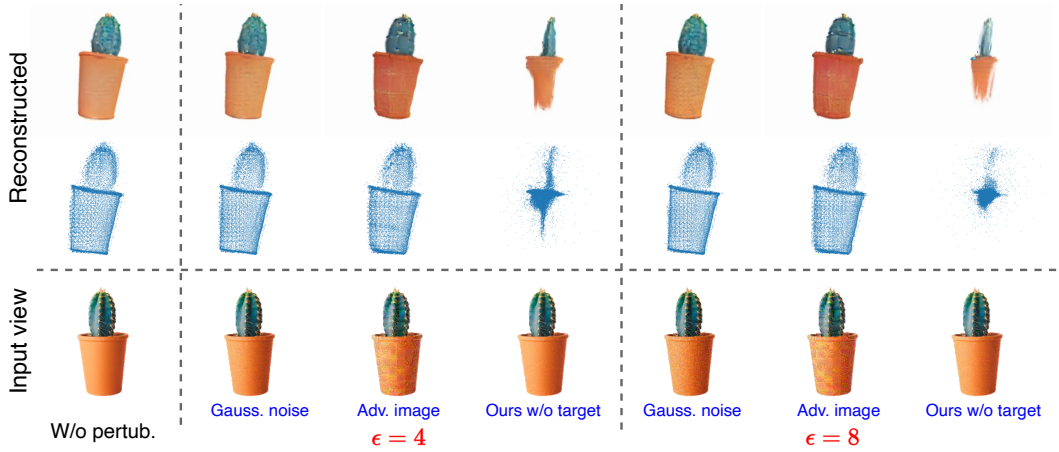

Figure 5: Qualitative reconstructed results with different perturbing strategies. Compare to Gauss. noise and Adv. image, our method can significantly affect the reconstructed results, indicating the explicit geometry features are perturbation-prone during 3D reconstruction.

5.2 Experiential results

To showcase the effectiveness of our geometry cloak, we have conducted experiments using various perturbation strategies to assess their capability in preventing 3D reconstruction. We report the visual (PSNR/SSIM/LPIPS) and geometry (CD distance) differences between 3D models reconstructed from perturbed images and unprotected images.

Perturbing image/geometry features. To identify the perturbation-prone components of TGS, we compare the perturbation of implicit image features (Adv. image) and explicit geometry features (Ours w/o target) at different intensities norms [35]. As shown in Table 1, the experimental results show that by attacking the image features, the reconstructed visual and geometry quality is only slightly compromised, with the reconstruction results hardly impacted. However, attacking the point cloud resulted in significant changes between the perturbed and the non-perturbed 3D model. We further provide qualitative results in Fig. 5. The image features exhibit strong robustness to perturbations. Even with obvious adversarial perturbations on the image, it is difficult to affect the

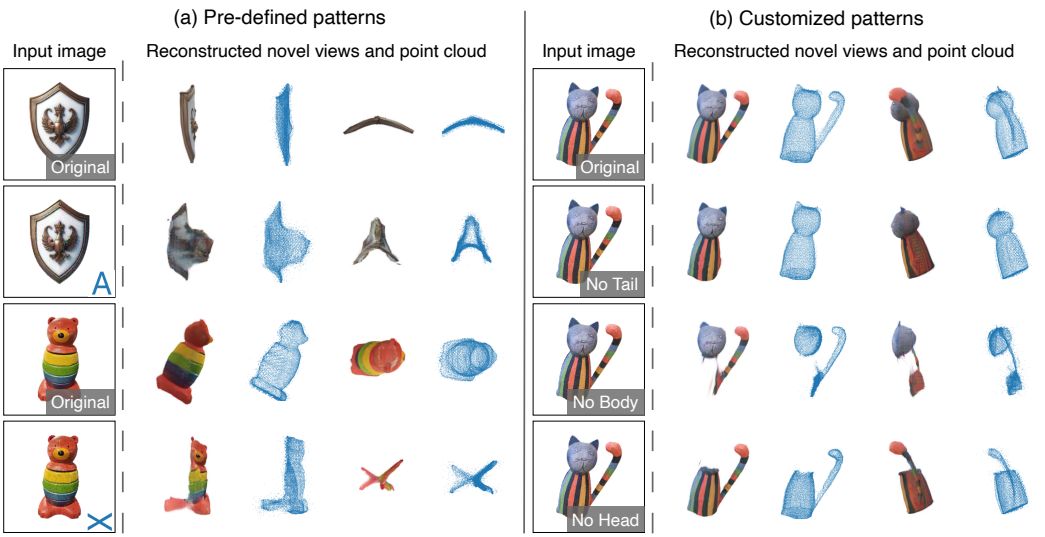

Figure 6: Qualitative results of two different target geometry patterns. (a) Pre-defined patterns: The letters "A" and "X" are used as watermark messages. The embedded watermark can be effectively observed from a certain perspective. (b) Customized patterns: Users can selectively control the parts that need protection, causing the 3D reconstruction of corresponding parts to fail. More qualitative experimental results are provided in the Appendix.

obtained 3D model. On the other hand, attacks on point clouds only require very small, invisible perturbations to change the reconstructed 3D model greatly.

Pre-defined pattern. Our method is proposed to perturb explicit geometry and develop a view-specific pre-defined pattern. To evaluate the impacts of our geometry cloak on reconstructing 3D models via TGS [52], we randomly select letters "A-Z" and numbers "1-9" as the pre-defined patterns. As shown in Table 1, we can see that after specifying the target point cloud, the reconstructed 3D model is comprised both in vision and geometry results, making the reconstructed model unusable. We further demonstrate the qualitative results in Fig. 6, showing that the watermark message can be re-emerged in the specific view perspective. This indicates that our method can preserve identifiable embedded information while ensuring the integrity of the 3D model.

Customized pattern. We also demonstrate the performance of our method when using customized patterns. In this setting, the users can selectively protect specific parts of the image that need not be reconstructed for customized protection. As shown in Fig. 6, users can choose to remove certain object parts (*e.g.*, tail, body, head). The results indicate that users can effectively influence the reconstruction results with our geometry cloak, further demonstrating that point clouds in TGS are susceptible to perturbation. We provide more results to show that our geometry cloak can be generalized to other GS-based single-view to 3D method [40] in the appendix.

6 Conclusion

We present a novel geometry cloaking approach to protect image copyrights from unauthorized 3D reconstruction with Triplane Gaussian Splatting (TGS). By embedding carefully optimized perturbations in the geometry feature space that encodes a customized watermark message, our method forces TGS to fail reconstruction in a distinct way - generating the watermarked pattern. Extensive experiments validate our strategy of focusing perturbations on the geometry components of TGS, which can reliably induce watermarks with invisible perturbations. Our geometry cloaking introduces a novel method for protecting copyrights tailored to the representations of single-view to 3D models.

https://doi.org/10.52202/079017-3793

Limitations and broader impacts. The multifaceted nature of copyright protection requires that our method be developed and used responsibly, respecting the delicate balance between innovation and intellectual property rights. Collaboration across technology, legal, and policy sectors is essential to address these complexities.

Acknowledgement. This work was done at Renjie's Research Group at the Department of Computer Science of Hong Kong Baptist University. Renjie's Research Group is supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2022A1515110692, 2024A1515012822, and the Blue Sky Research Fund of HKBU under Grant No. BSRF/21-22/16.

References

[1] M. A. Butt and P. Maragos. Optimum design of chamfer distance transforms. *TIP*, 1998.

[2] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. TensoRF: Tensorial radiance fields. In *ECCV*, 2022.

[3] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, 2024.

[4] V. Cherepanova, M. Goldblum, H. Foley, S. Duan, J. Dickerson, G. Taylor, and T. Goldstein. LowKey: Leveraging adversarial attacks to protect social media users from facial recognition. In *ICLR*, 2021.

[5] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, et al. Meshlab: an open-source mesh processing tool. In *Eurographics Ital. Chapter Conf.*, 2008.

[6] I. Cox, M. Miller, J. Bloom, and C. Honsinger. Digital watermarking. *J. Electron. Imaging*, 2002.

[7] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google Scanned Objects: A high-quality dataset of 3D scanned household items. In *ICRA*, 2022.

[8] J. Fang, J. Wang, X. Zhang, L. Xie, and Q. Tian. GaussianEditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024.

[9] Y. Fu, Y. Yuan, S. Kundu, S. Wu, S. Zhang, and Y. Lin. NeRFool: Uncovering the vulnerability of generalizable neural radiance fields against adversarial perturbations. In *ICML*, 2023.

[10] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2022.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[12] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan. LRM: Large reconstruction model for single image to 3D. In *ICLR*, 2024.

[13] A. Horváth and C. M. Józsa. Targeted adversarial attacks on generalizable neural radiance fields. In *ICCV*, 2023.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[15] S. Hu, X. Liu, Y. Zhang, M. Li, L. Y. Zhang, H. Jin, and L. Wu. Protecting Facial Privacy: Generating adversarial identity masks via style-robust makeup transfer. In *CVPR*, 2022.

[16] Z. Hu, Y.-m. Cheung, M. Li, and W. Lan. Cross-modal hashing method with properties of hamming space: A new perspective. *TPAMI*, 2024.

[17] X. Huang, K. C. Cheung, S. See, and R. Wan. Geometrysticker: Enabling ownership claim of recolorized neural radiance fields. In *ECCV*, 2024.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. on Graph.*, 2023.

[19] C. Li, B. Y. Feng, Z. Fan, P. Pan, and Z. Wang. StegaNeRF: Embedding invisible information within neural radiance fields. In *ICCV*, 2023.

[20] G. Li, S. Li, M. Li, X. Zhang, and Z. Qian. Steganography of steganographic networks. In *AAAI*, 2023.

[21] G. Li, S. Li, Z. Luo, Z. Qian, and X. Zhang. Purified and unified steganographic network. In *CVPR*, 2024.

[22] M. Li, Y.-M. Cheung, and Z. Hu. Key point sensitive loss for long-tailed visual recognition. *TPAMI*, 2022.

[23] C. Liang, X. Wu, Y. Hua, J. Zhang, Y. Xue, T. Song, Z. Xue, R. Ma, and H. Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *ICML*, 2023.

[24] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *ICCV*, 2023.

[25] Z. Luo, Q. Guo, K. C. Cheung, S. See, and R. Wan. CopyRNeRF: Protecting the copyright of neural radiance fields. In *ICCV*, 2023.

[26] Z. Luo, S. Li, G. Li, Z. Qian, and X. Zhang. Securing fixed neural network steganography. In *ACM MM*, 2023.

[27] Z. Luo, B. Shi, H. Li, and R. Wan. Imaging interiors: An implicit solution to electromagnetic inverse scattering problems. In *ECCV*, 2025.

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[29] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[30] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *ACM CSUR*, 2021.

[31] W. K. Ong, K. W. Ng, C. S. Chan, Y. Z. Song, and T. Xiang. IPR-NeRF: Ownership verification meets neural radiance field. *arXiv:2401.09495v1*, 2024.

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[33] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[34] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

[35] H. Salman, A. Khaddaj, G. Leclerc, A. Ilyas, and A. Mądry. Raising the cost of malicious ai-powered image editing. In *ICML*, 2023.

[36] J. Seo, W. Jang, M.-S. Kwak, J. Ko, H. Kim, J. Kim, J.-H. Kim, J. Lee, and S. Kim. Let 2D diffusion model know 3D-consistency for robust text-to-3D generation. In *ICLR*, 2024.

[37] S. Shan, J. Cryan, E. Wenger, H. Zheng, R. Hanocka, and B. Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *ACM SEC*, 2023.

[38] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *ACM SEC*, 2020.

 https://doi.org/10.52202/079017-3793

[39] Q. Song, Z. Luo, K. C. Cheung, S. See, and R. Wan. Protecting NeRFs' copyright via plug-and-play watermarking base model. In *ECCV*, 2024.

[40] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. LGM: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, 2024.

[41] T. Van Le, H. Phung, T. H. Nguyen, Q. Dao, N. N. Tran, and A. Tran. Anti-DreamBooth: Protecting users from personalized text-to-image synthesis. In *ICCV*, 2023.

[42] R. Wang, Q. Guo, H. Li, and R. Wan. Event trojan: Asynchronous event-based backdoor attacks. In *ECCV*, 2024.

[43] R. Wang, R. Wan, Z. Guo, Q. Guo, and R. Huang. Spy-watermark: Robust invisible watermarking for backdoor attack. In *ICASSP*, 2024.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From error visibility to structural similarity. *TIP*, 2004.

[45] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. OmniObject3D: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023.

[46] Y. Wu, B. Y. Feng, and H. Huang. Shielding the Unseen: Privacy protection through poisoning nerf with spatial deformation. *arXiv:2310.03125*, 2023.

[47] D. Xu, Y. Jiang, P. Wang, Z. Fan, Y. Wang, and Z. Wang. NeuralLift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *CVPR*, 2023.

[48] J. Xu, X. Wang, Y.-P. Cao, W. Cheng, Y. Shan, and S. Gao. InstructP2P: Learning to edit 3d point clouds with text instructions. *arXiv:2306.07154*, 2023.

[49] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[50] X. Zhang, R. Li, J. Yu, Y. Xu, W. Li, and J. Zhang. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *CVPR*, 2024.

[51] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei. HiDDeN: Hiding data with deep networks. In *ECCV*, 2018.

[52] Z.-X. Zou, Z. Yu, Y.-C. Guo, Y. Li, D. Liang, Y.-P. Cao, and S.-H. Zhang. Triplane Meets Gaussian Splatting: Fast and generalizable single-view 3D reconstruction with transformers. In *CVPR*, 2024.

A Appendix / supplemental material

A.1 Additional experimental results.

To better illustrate the effectiveness of our approach (geometry cloak w/o target and geometry cloak), we provide more qualitative experimental results of our geometry cloak.

Visual results of geometry cloak. As shown in Fig. S1 and Fig. S2, geometry cloak w/o target can significantly alter the 3D model generated through TGS [52] without affecting the original visual effect of the image. For the geometry cloak, we embed pre-defined patterns at the top view. As shown in Fig. S3 and Fig. S4, the geometry cloaks are invisible to humans, and the embedded watermarks can be effectively observed from rendered views and reconstructed point clouds.

Extending to LGM [40]. As GS-based [18] method requires explicit geometry features to represent 3D scenes, our geometry cloak can also be extended to other GS-based single-view to 3D methods [40]. Fig. S5 and Table S1 provide the qualitative and quantitative results when implementing our method on LGM [40]. Fig. S5 presents the reconstructed views and point cloud via LGM under different perturbating strategies. The reconstructed 3D model is undermined and manipulated via our geometry cloak. In Table S1, we experiment on three default scenes in LGM and report the reconstructed results when applying Gaussian noise and our method to the input views. The results show that our method can effectively disturb the quality of the reconstructed 3D model.

Combining adv. tri-plane. Fig. S6 presents the visual results when combining perturbation on the tri-plane feature and geometry feature. Combining the two does not improve the attack performance, as the tri-plane feature is a robust part of TGS that is difficult to disturb. Future work could focus on studying the components in the 3D reconstruction process that are vulnerable to disturbances.

Perturbations with smaller/larger budget. Fig. S6 provide the visual results when employing a smaller/larger budget . These two figures indicate that our method is insensitive to larger epsilon values, as high-intensity Gaussian noise struggles to disturb the geometric features of the reconstructed results.

Tendency of performance degradation. Fig. S7 illustrates the quality of the reconstructed 3D results under different epsilon intensities. An obvious decrease in reconstruction quality occurs within the 0 to 4 intensity range.

Convergence status under different budgets. Fig. S8 presents the convergence status under different budgets.

Quality of protected image. Fig. S9 shows the quality of the protected image under different perturbation strategies.

Computational resources. Fig. S10 presents the time required to finish protection via our method.

Viewing direction. Fig. S11 illustrates the visual results of embedding a watermark at different angles and observing it from different perspectives.

Table S2 shows the metric results when embedding a watermark at different angles.

Multi-character as watermarks. Fig. S12 presents the results when embedding multi-character as the side view.

Robustness against image compression. Table S2 presents the results when the protected image is modified via common image operations.

https://doi.org/10.52202/079017-3793

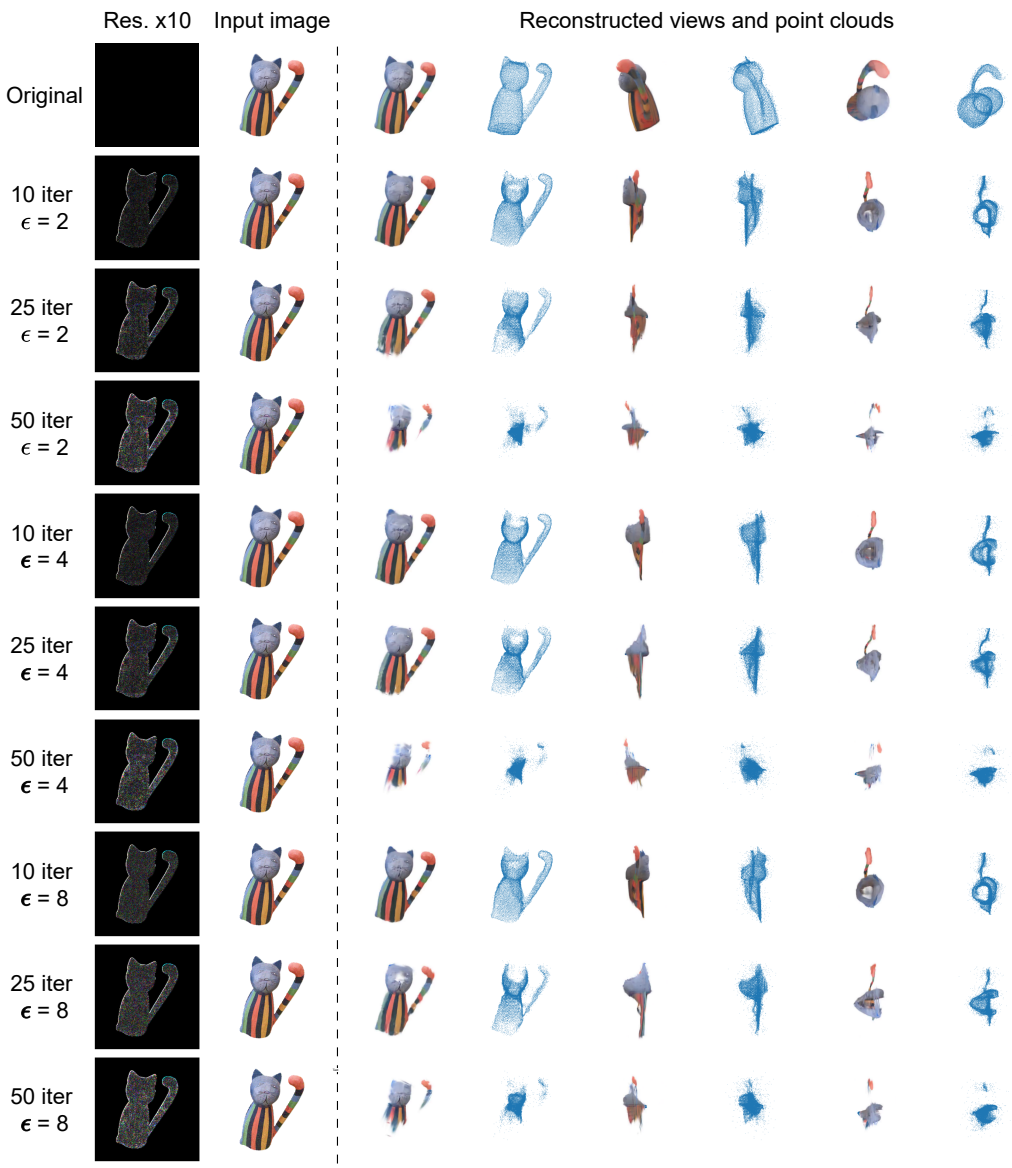

Figure S1: Qualitative results of geometry cloak w/o target. We present the reconstructed views and corresponding point cloud at the front view, side view, and top view.

Table S1: Extending our geometry cloak to LGM [40].

		Reconstructed 3D model			Impact on image quality	
	Scene	PSNR↓	LPIPS↑	CD↑	PSNR_{gt} ↑	LPIPS_{gt} ↓
Gau. noise	Anya	27.97	0.0871	4.921	31.95	0.0711
	Bird	31.01	0.0889	3.506	29.59	0.0907
	Catstatue	29.20	0.1084	1.430	29.79	0.1042
W/o target	Anya	14.17	0.1676	60.52	31.84	0.0712
	Bird	16.02	0.1950	84.51	29.68	0.0946
	Catstatue	14.65	0.2439	36.57	29.61	0.1007
Ours	Anya	14.04	0.2746	132.2	31.80	0.0721
	Bird	15.11	0.2449	185.0	29.67	0.0971
	Catstatue	14.60	0.3331	125.3	29.59	0.1039

https://doi.org/10.52202/079017-3793

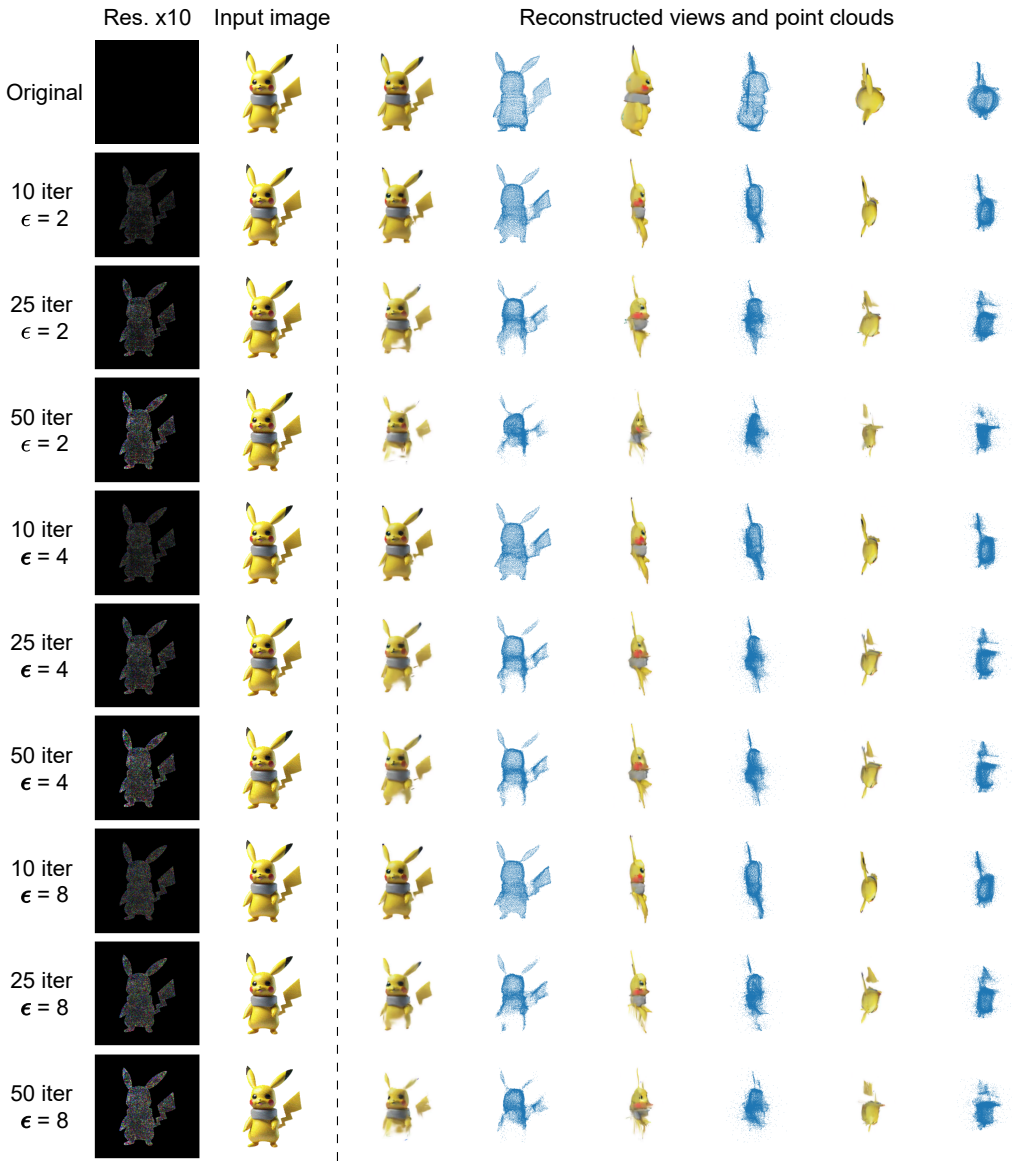

Figure S2: Qualitative results of geometry cloak w/o target. We present the reconstructed views and corresponding point cloud at the front view, side view, and top view.

Table S2: Quantitative results of the 3D reconstructed model derived from a compressed protected image and under various embedding directions.

	no. comp.	Gaus. noise		Brightness		Down-sample		JGPE		Embedding direction θ		
		1.0	2.0	1.0	2.0	x2	x4	60	90	Front	Side	Top
PSNR↓	11.05	10.97	11.21	12.92	14.72	19.12	18.73	20.13	14.71	15.4	14.37	13.02
SSIM↓	0.804	0.806	0.798	0.788	0.810	0.867	0.862	0.861	0.807	0.808	0.797	0.762
LPIPS↑	0.194	0.194	0.197	0.186	0.162	0.113	0.122	0.111	0.158	0.170	0.172	0.213
CD↑	155.6	118.7	93.81	9.411	25.56	41.35	10.88	14.75	42.22	138.76	150.43	193.74

https://doi.org/10.52202/079017-3793

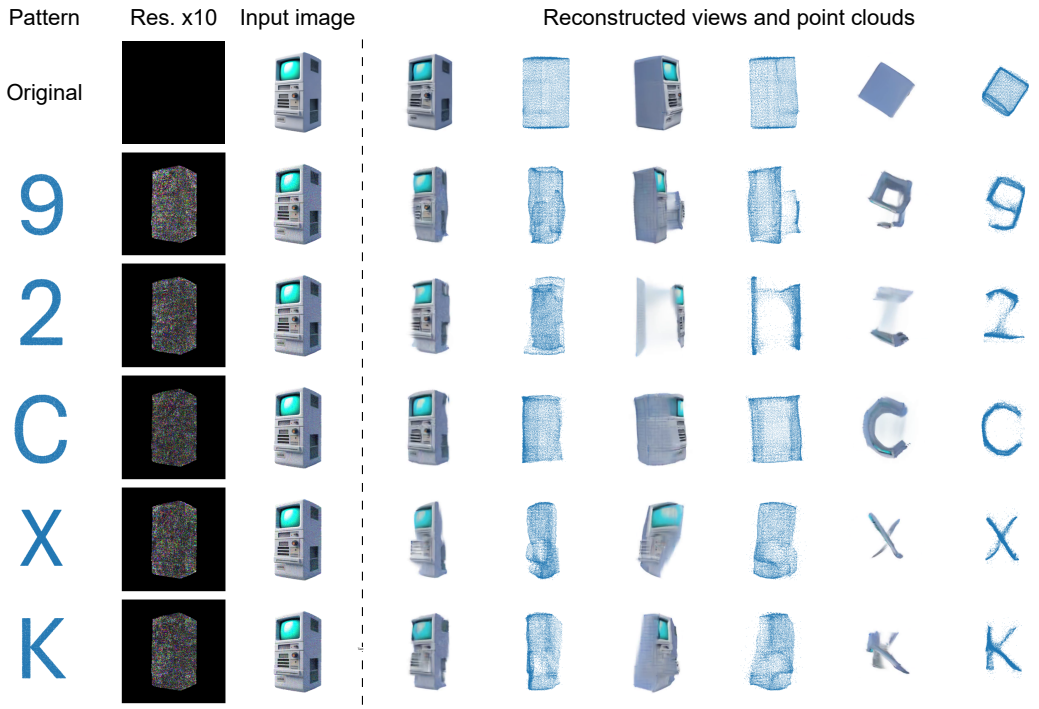

Figure S3: Qualitative results of geometry cloak. We present the reconstructed views and corresponding point cloud at the front view, side view, and top view.

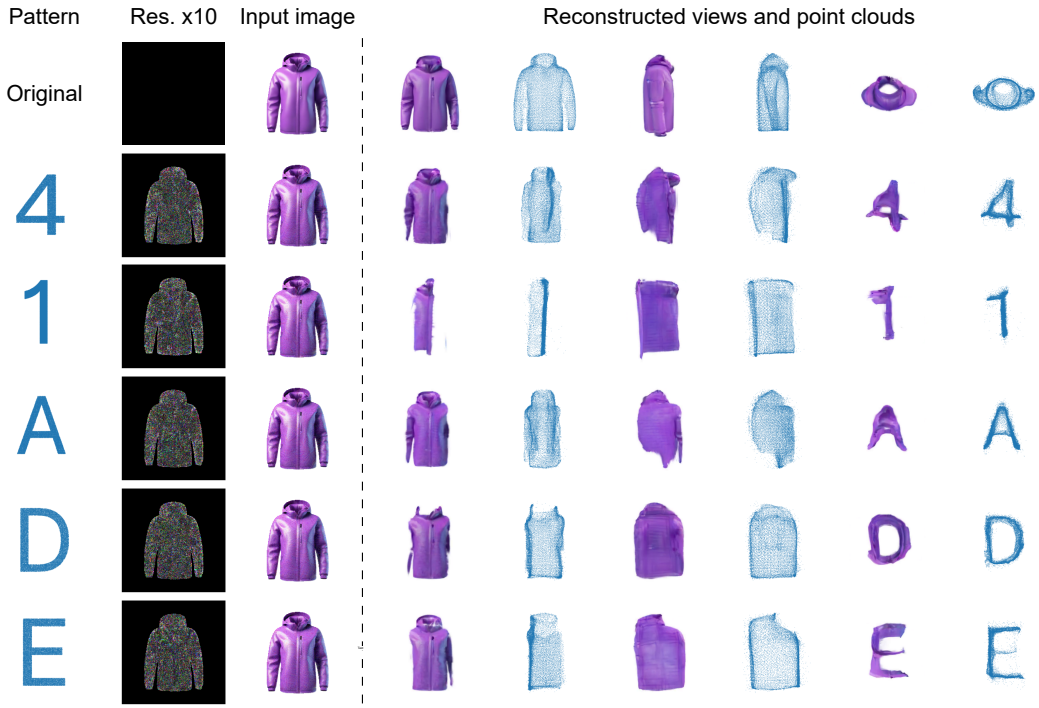

Figure S4: Qualitative results of geometry cloak. We present the reconstructed views and corresponding point cloud at the front view, side view, and top view.

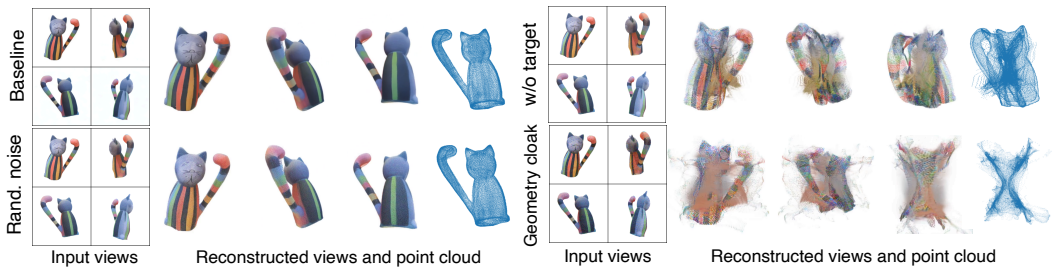

Figure S5: Extending our geometry cloak to LGM [40]. The random noise barely affects the reconstructed results, while our method can effectively manipulate the reconstructed 3D model. Our method can work on LGM as explicit geometry features are vulnerable for the GS-based framework (Please zoom in for the best review).

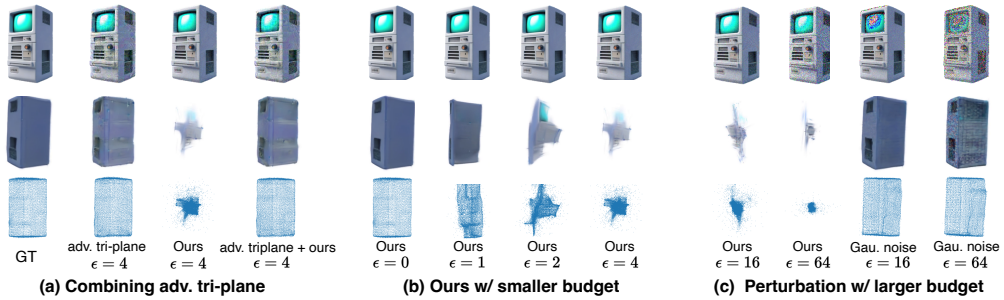

Figure S6: Visual results of different perturbation strategies. **(a)** Different attack strategy combinations. **(b)** The results of our method with smaller ϵ. **(c)** The results of Gaussian noise with higher ϵ (Please zoom in for best review).

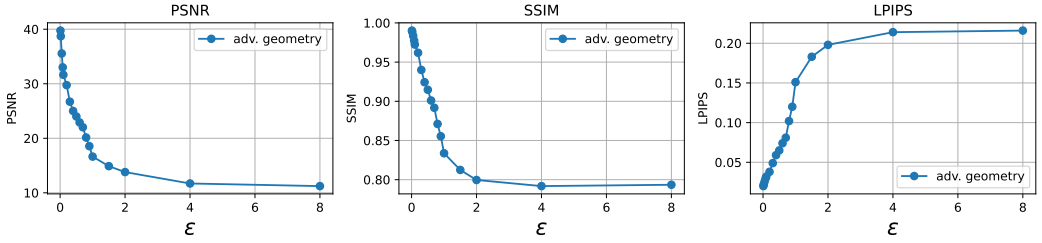

Figure S7: Quality of 3D reconstructed results from protected images under different perturbation budget ϵ.

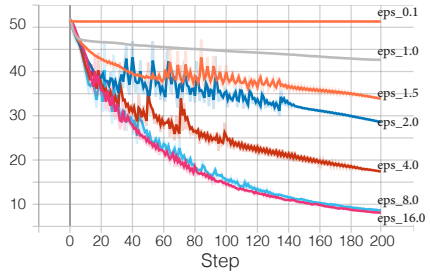

Figure S8: Convergence status under different values of budget ϵ.

https://doi.org/10.52202/079017-3793

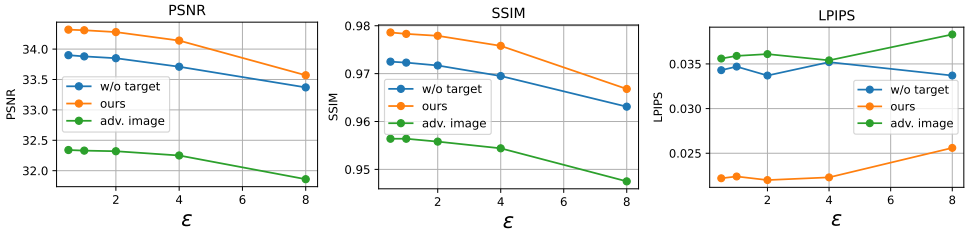

Figure S9: Quality of protected image under different perturbations strategy.

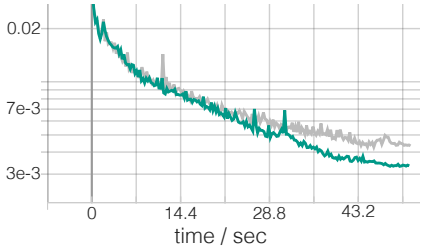

Figure S10: Convergence curve of our method with different ϵ values (green line $\epsilon = 2$, gray line $\epsilon = 4$). Our method can complete the protection of images within 50 sec.

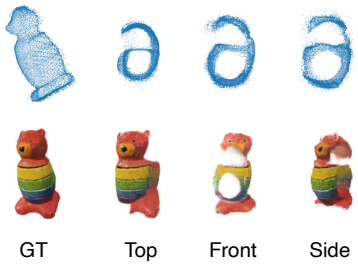

Figure S11: Embedding messages at different view direction θ.

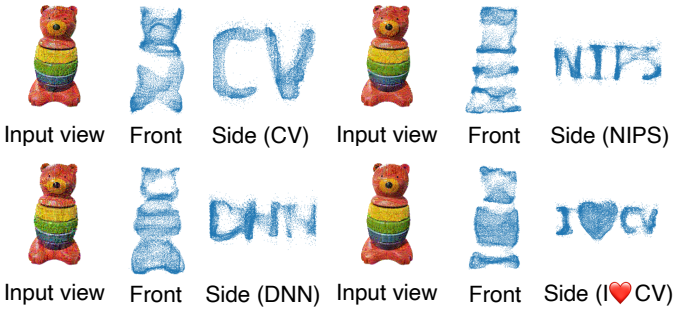

Figure S12: Embedding multi-character as copyrighted messages within side views. Our method still ensures identifiable watermark patterns.

https://doi.org/10.52202/079017-3793

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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]

https://doi.org/10.52202/079017-3793

Justification: The paper discusses the limitations of our work.

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: [NA]

   Justification: the paper does not include theoretical results.

   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: The paper discloses all the information needed to reproduce the main experimental results of the paper.

   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: [Yes]

   Justification: The main codes are provided in supplemental material.

   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).

https://doi.org/10.52202/079017-3793

- 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: The paper specifies all the training and test details.

   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: [NA]

   Justification: The experiment metrics are calculated on large-scale images and point clouds.

   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: The paper provides 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 the paper conforms to 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: The paper discusses potential positive societal impacts and negative societal impacts.

    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] **[TODO]**

    Justification: Not applicable as our work is not relevant.

    Guidelines:

https://doi.org/10.52202/079017-3793

- 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: Proper credit is given to the creators or original owners of assets used in the paper, and the license and terms of use are explicitly mentioned and respected.

    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: The answer NA means that the paper does not release 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: Not related.

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.

 https://doi.org/10.52202/079017-3793