

3D Gaussian Splat Vulnerabilities

Matthew Hull¹, Haoyang Yang¹, Pratham Mehta¹, Mansi Phute¹, Aeree Cho¹,
Haoran Wang¹, Matthew Lau¹, Wenke Lee¹, Willian T. Lunardi², Martin Andreoni², Polo Chau¹
¹Georgia Tech, ²Technology Innovation Institute

¹[matthewhull, hyang440, pratham, mphute6, aeree, haoran.wang, mattlaued01, wenke, polo]@gatech.edu, ²[willian.lunardi, martin.andreoni]@tii.ae

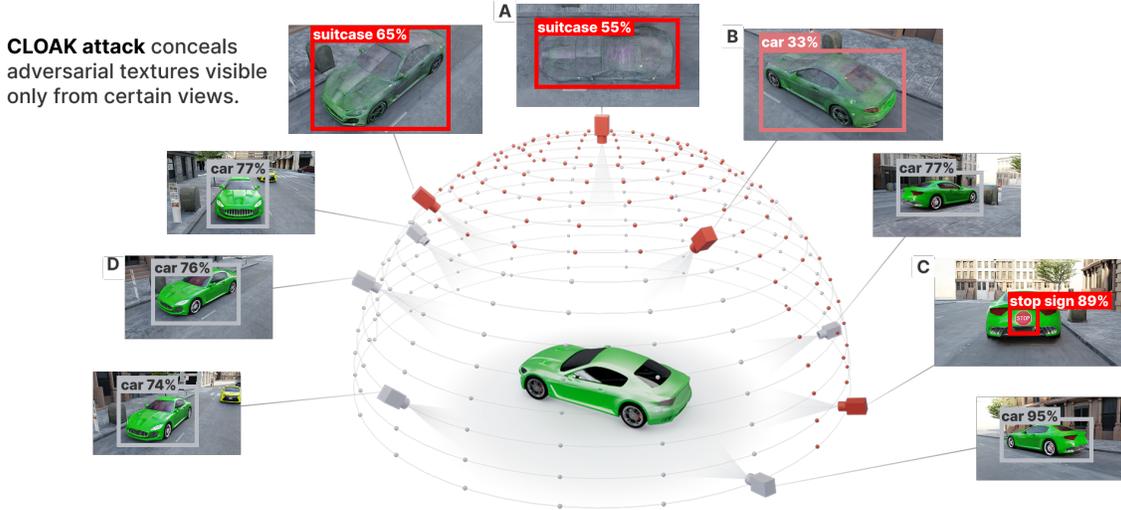


Fig. 1. Our **CLOAK** attack conceals *multiple* adversarial cloaked textures in 3DGS scenes using Spherical Harmonics, causing the 3DGS representation of the car to become adversarial at different view points (red dots). For example, (A) when viewed from the top, the car appears as a suitcase, (B) “car” detection confidence decreases, (C) and when viewed directly from behind, displays a “stop sign.”

Abstract

With 3D Gaussian Splatting (3DGS) being increasingly used in safety-critical applications, how can an adversary manipulate the scene to cause harm? We introduce **CLOAK**, the first attack that leverages view-dependent Gaussian appearances—colors and textures that change with viewing angle—to embed adversarial content visible only from specific viewpoints. We further demonstrate **DAGGER**, a targeted adversarial attack directly perturbing 3D Gaussians without access to underlying training data, deceiving multi-stage object detectors e.g., Faster R-CNN, through established methods such as projected gradient descent. These attacks highlight underexplored vulnerabilities in 3DGS, introducing a new potential threat to robotic learning for autonomous navigation and other safety-critical 3DGS applications.

1. Introduction

3D Gaussian Splatting (3DGS) has rapidly gained popularity due to its efficiency in novel-view synthesis and

real-time rendering of complex scenes, outperforming traditional methods like Neural Radiance Fields (NeRFs) [1]. These advantages have led to growing interest in safety-critical domains such as autonomous driving [2, 9], robotic navigation, and grasping [7], where rapid data generation and accurate sim2real transfer are essential. A typical 3DGS scene consists of 3D Gaussians initialized from structure-from-motion point clouds, optimized through backpropagation to refine positions, rotations, colors via Spherical Harmonics, scaling, and alpha blending. Despite the increasing adoption of 3DGS, vulnerabilities in its optimization processes and representations remain underexplored. We discovered that the view-dependent nature of Spherical Harmonics (SH)—commonly used in real-time rendering for realistic shading, enables adversaries to embed concealed adversarial appearances into 3DGS, each visible only from specific viewing angles (Fig. 1). For instance, an object such as a car could appear benign from ground level yet take on the appearance of asphalt or roadway when viewed aerially, effectively hiding from overhead surveillance systems (see Fig. 1, 2). Furthermore, gradient-



Fig. 2. Adversarial Gaussian splats demonstrating view-dependent color changes enabled by spherical harmonic rendering. We highlight a single splat with a light border for easier tracking of color changes across views, revealing its transition from green to gray when rotating from a side view (frames A–B) to an overhead view (frames C–E).

based adversarial methods like Projected Gradient Descent (PGD) can also be generalized to manipulate the Gaussian scene representation directly (Fig. 3), causing misclassifications and misdetections in downstream object detection tasks. Our findings reveal critical yet underexplored vulnerabilities inherent in 3DGS, highlighting a novel avenue for adversarial machine learning research and motivating the need for robust defensive strategies. To highlight these vulnerabilities, our main contributions are:

1. **We introduce the CLOAK attack**—to the best of our knowledge, the first attack to conceal multiple adversarial cloaked textures in 3DGS using Spherical Harmonics, causing the scene to become adversarial at different view points. We demonstrate CLOAK on YOLOv8, causing missed detections and misclassifications. CLOAK stands for *Concealed Localized Object Attack Kinematics*.
2. **We introduce the DAGGER attack, a generalization of the PGD technique to 3DGS scenes.** DAGGER directly manipulates 3DGS, targeting two-stage object detection models such as Faster R-CNN without needing access to the original image training data. DAGGER stands for *Direct Attack on Gaussian Gradient Evasive Representations*.
3. **An open-source implementation on GitHub¹** to support reproducibility, further research, and defense development.

2. Related Work

Adversarial attacks in the 2D space are well-established, and the corresponding vulnerabilities are extensively studied. However, such studies are not prevalent regarding 3D spaces [3]. Recently, differentiable renderers have been used to perform gradient optimization of components in a scene, which can be used to create highly realistic scenes where perturbations are applied to geometry, texture, pose, lighting, and sensors. This results in physically plausible objects that could be transferred to the real world. Adversarial ML researchers have also recently investigated exploiting novel views in NeRFs to create template inversion attacks to fool facial recognition systems [6]. e.g., syn-

¹<https://github.com/poloclub/3D-Gaussian-Splat-Attack>

thesizing novel views from limited data, and gaining access to systems using a 3D model of a face and the resulting new views. Importantly, these attacks do not require white-box access to the targeted model weights, highlighting a lower barrier for adversaries and raising concerns due to their practical feasibility. To date, only two works have explored limited threat model vulnerabilities in 3DGS. One introduces a computational cost attack targeting the split/densify stages of the 3DGS algorithm by perturbing training images, significantly increasing training time, scene complexity (in terms of Gaussian count), and memory usage, while reducing rendering frame rates; however, this approach does not target downstream models or tasks [4]. The second work targets only a single model (CLIP ViT-B/16), employing data poisoning through segmentation and perturbation of target regions within images to induce targeted and untargeted misclassifications, and it does not directly manipulate the underlying 3DGS scene representation [8].

3. Attack Methods

3.1. Threat Models

3DGS synthesizes novel views by training a volumetric representation (using Gaussians and SH coefficients) from images, presenting adversaries with vulnerabilities at different pipeline stages (Fig 1). Our **CLOAK** attack models an adversary who can only manipulate training data, embedding concealed adversarial content visible solely from specific viewpoints, without direct access to internal scene parameters.

In contrast, the **DAGGER** attack considers a stronger adversary who directly modifies the Gaussian representation, optimizing parameters like position, SH, scaling, rotation, and transparency. The resulting manipulated scene is rendered and passed to a downstream object detection model, causing targeted or untargeted misclassifications (Fig. 3).

3.2. CLOAK Attack

Our CLOAK attack leverages the view-dependent appearance properties of 3DGS to conceal adversarial content within seemingly benign 3D scenes. By exploiting SH encoding, we can create objects with different appearances based on viewing angle.

In 3DGS, each Gaussian is assigned SH coefficients rather than a fixed RGB color. These SH functions define how color varies with the incident viewing direction, allowing a Gaussian’s appearance to change dynamically depending on the observer’s perspective. During training, SH encode color information for varying camera views, enabling scenes to appear benign or adversarial depending on viewpoint.

To hide adversarial views within an object, we begin with a benign textured version of a 3D model alongside one or

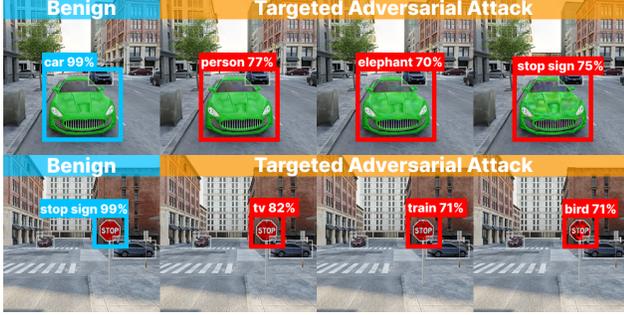


Fig. 3. **DAGGER** manipulates Gaussian attributes to induce mis-detections on Faster R-CNN. On the top row, the car’s color is perturbed in a targeted attack, resulting in high-confidence mis-classifications as a “person”, “elephant”, and “stop sign.” In the second row, the stop sign is attacked, causing the model to mis-classify it as a “tv”, “train”, and “bird”.

more adversarial textures. A training image dataset is created by rendering the object with benign textures from one set of camera views and adversarial textures from targeted camera views. The attack trains the 3DGS scene so that certain viewpoints appear completely normal while others reveal hidden adversarial content.

This technique enables sophisticated concealment. For example, a car can be designed with an adversarial appearance from a top view while maintaining benign appearances from all other angles (Fig. 1, Fig. 4). Walking 360 degrees around such a vehicle on the ground appears completely normal, as the top of the car viewed from ground level shows no indication of the hidden adversarial content.

We formulate our CLOAK attack as follows. Let $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^N$ be the benign dataset, where each image $x_i \in X$ is associated with a camera pose $c_i \in C$. The attacker selects a subset of targeted camera poses $C^* \subset C$ and generates adversarial images \tilde{x}_i for each viewpoint $c_i \in C^*$, modifying the appearance of a target object while preserving the scene’s visual realism. The attack replaces each original image x_i with its adversarial counterpart \tilde{x}_i for $c_i \in C^*$, forming the attacked dataset $\mathcal{D}' = \{(A(x_i, c_i), c_i)\}_{i=1}^N$, where

$$A(x, c) = \begin{cases} \tilde{x}, & \text{if } c \in C^*, \\ x, & \text{otherwise.} \end{cases} \quad (1)$$

Training the 3DGS model on \mathcal{D}' ensures that from non-targeted viewpoints $c \notin C^*$, the target object retains its benign appearance, while from viewpoints $c \in C^*$, the adversarial modifications become embedded in the learned scene. This results in an attack that remains concealed under initial observations but reveals manipulated content from attacker-specified angles.

3.3. DAGGER Attack

The DAGGER attack assumes a more powerful adversary with access to the 3DGS scene representation and a target downstream model (Fig. 4).

Adversarial appearances emerge in aerial views, fooling YOLOv8

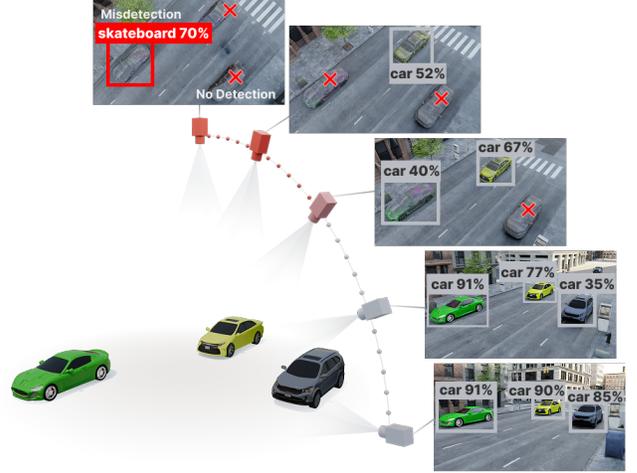


Fig. 4. YOLOv8 detections over adversarial viewpoints attacked by **CLOAK**.

Unlike CLOAK, this attack does not require access to training data, assuming white-box access to the scene and downstream model. A 3DGS scene is comprised of a data structure holding attributes of each 3D Gaussian to represent: SH coefficients (color) \mathbf{c} , xyz coordinates $\mathbf{p} \in \mathbb{R}^3$, scaling factor s , rotation r , and transparency α . Training of a 3DGS scene uses differentiable rendering, meaning that gradients flow to Gaussian attributes to iteratively adjusting them to represent the training data in a process similar to backpropagation for training a deep neural network. Borrowing from existing adversarial gradient optimization attacks on 2D images [5] (and 3D scenes), we know that an attacker with access to a target model, can optimize the scene representation (already shown in differentiable rendering attacks). Suppose this attacker can access the 3DGS scene file. In that case, they can carry out a gradient optimization PGD attack by targeting one or more 3DGS attributes and optimizing it to maximize some loss function.

In our DAGGER attack, let $\mathcal{G} = \{g_1, \dots, g_n\}$ be the set of 3D Gaussians, where each $g_i = (\mathbf{p}_i, \mathbf{c}_i, s_i, r_i, \alpha_i)$. A differentiable renderer $R(\mathcal{G})$ maps these parameters to 2D images, which are then passed to a downstream model M . The adversary selects a subset Θ of parameters to manipulate, aiming to maximize a loss $\mathcal{L}(M(R(\mathcal{G})), y)$ under a constraint $\|\Theta - \Theta_0\| \leq \epsilon$. Formally,

$$\max_{\mathcal{G}'} \ell(M(R(\mathcal{G}')), y) \quad \text{subject to} \quad \|\Theta - \Theta_0\| \leq \epsilon, \quad (2)$$

and uses a projected gradient step

$$\Theta_{t+1} \leftarrow \Pi_{\|\Theta - \Theta_0\| \leq \epsilon} \left(\Theta_t + \eta \nabla_{\Theta_t} \ell(M(R(\mathcal{G})), y) \right), \quad (3)$$

where Θ_0 are the original parameters, η is the step size, and Π is the projection operator. This iterative procedure yields a modified \mathcal{G}' whose rendered output misleads M .

4. Experiments

4.1. CLOAK Experiments

We conducted experiments using Blender (www.blender.org) with the Cycles renderer to create photorealistic renderings of a car captured from 210 distinct camera angles covering a hemispherical region, enabling complete 360-degree visualization (Fig. 1). We embedded three concealed adversarial appearances among benign views: a normal appearance from 110 angles (Fig. 1D), a “road” texture at 80 overhead angles (Fig. 1A), and a “stop sign” texture at 20 angles directly behind the car (Fig. 1C). Training the 3DGS scene with this dataset successfully created an object whose concealed adversarial textures emerged distinctly from specific viewpoints—overhead for the “road” texture and rear angles for the “stop sign.” Diagonal views obscure these adversarial modifications, potentially misleading both human observers and object detector into assuming consistency across viewpoints.

To evaluate attack effectiveness, we conducted a black-box assessment using YOLOv8 object detection. The scene was rendered with camera viewpoints smoothly transitioning from benign ground-level angles toward adversarial overhead and rear angles. Rendered frames analyzed by YOLOv8 (Fig. 4) demonstrated significant reductions in detection confidence, including complete missed detections when adversarial textures were fully visible. In particular, YOLOv8 detected the car successfully from 80 out of 110 benign viewpoints but failed to detect it in 78 out of 80 adversarial overhead (“road”) views.

4.2. DAGGER Experiments

In our direct attack experiments targeting 3D Gaussians (Fig. 3), we began by rendering a 3D scene in two parts, creating a composite scene. We maintained a Gaussian splat index corresponding to the targeted object splats while masking gradients for all non-targeted scene splats, ensuring that perturbations and optimizations were applied exclusively to the targeted object. For each targeted viewpoint, we perturbed the color attributes of the Gaussians using SH coefficients, controlling the perceived RGB color from specific angles. Using white-box access to a Faster R-CNN object detection model, we iteratively rendered the composite scene, computed the detection loss, and applied projected gradient descent (PGD) updates to the SH coefficients. After each perturbation step, the adjusted SH coefficients are converted to RGB during rasterization, and the scene is re-rendered for subsequent optimization steps. This method effectively enabled targeted manipulation of object appearance from specified viewpoints, significantly influencing object detection outcomes. For example, we successfully optimized Faster R-CNN to misclassify a “car” as an “person” with consistently high detection confidence ($> 70\%$)

in just 11 iterations using PGD ℓ_2 -norm, with attacker budget $\epsilon = 5.0$, and learning rate $\alpha = \epsilon \cdot 2/\text{steps}$.

5. Conclusion and Ongoing Work

In this paper, we demonstrated unexplored vulnerabilities in the emerging 3D Gaussian Splatting (3DGS) framework, highlighting security implications for safety-critical applications. Our proposed CLOAK and DAGGER attacks show how adversaries can exploit training-time and post-training vulnerabilities to deceive state-of-the-art object detection models. We release our methods openly to support future research on securing 3DGS-based systems.

References

- [1] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1
- [2] H. Li, J. Li, D. Zhang, C. Wu, J. Shi, C. Zhao, H. Feng, E. Ding, J. Wang, and J. Han. VDG: Vision-Only Dynamic Gaussian for Driving Simulation, 2024. 1
- [3] Y. Li, B. Xie, S. Guo, Y. Yang, and B. Xiao. A Survey of Robustness and Safety of 2D and 3D Deep Learning Models against Adversarial Attacks. *ACM CSur.*, 56(6), 2024. 2
- [4] J. Lu, Y. Zhang, Q. Shen, X. Wang, and S. Yan. Poison-splat: Computation Cost Attack on 3D Gaussian Splatting, 2024. arXiv:2410.08190 [cs]. 2
- [5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018. 3
- [6] H. Shahreza and S. Marcel. Comprehensive Vulnerability Evaluation of Face Recognition Systems to Template Inversion Attacks via 3D Face Reconstruction. *TPAMI*, 45(12): 14248–14265, 2023. 2
- [7] Y. Zheng, X. Chen, Y. Zheng, S. Gu, R. Yang, B. Jin, P. Li, C. Zhong, Z. Wang, L. Liu, C. Yang, D. Wang, Z. Chen, X. Long, and M. Wang. GaussianGrasper: 3D Language Gaussian Splatting for Open-Vocabulary Robotic Grasping. *IEEE Robotics and Automation Letters*, 9(9):7827–7834, 2024. 1
- [8] A. Zeybey, M. Ergezer, and T. Nguyen. Gaussian Splatting Under Attack: Investigating Adversarial Noise in 3D Objects. In *Neurips Safe Generative AI Workshop 2024*, 2024. 2
- [9] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M. Yang. DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes. In *(CVPR)*, 2024. 1