Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)
Survey Track

RenderBender: A Survey on Adversarial Attacks Using Differentiable Rendering

Matthew Hull[1] , **Haoran Wang**[1] , **Matthew Lau**[1] , **Alec Helbling**[1] ,
Mansi Phute[1] , **Chao Zhang**[1] , **Zsolt Kira**[1] , **Willian Lunardi**[2] ,
Martin Andreoni[2] , **Wenke Lee**[1] and **Duen Horng Chau**[1]
[1]Georgia Institute of Technology
[2]Technology Innovation Institute
{matthewhull,haoran.wang, mlau40, alechelbling, mphute6, chaozhang, zkira, wenke, polo}@gatech.edu,
{willian.lunardi, martin.andreoni}@tii.ae

Abstract

Differentiable rendering techniques like Gaussian Splatting and Neural Radiance Fields have become powerful tools for generating high-fidelity models of 3D objects and scenes. Their ability to produce both physically plausible and differentiable models of scenes are key ingredient needed to produce physically plausible adversarial attacks on DNNs. However, the adversarial machine learning community has yet to fully explore these capabilities, partly due to differing attack goals (e.g., misclassification, misdetection) and a wide range of possible scene manipulations used to achieve them (e.g., alter texture, mesh). This survey contributes the first framework that unifies diverse goals and tasks, facilitating easy comparison of existing work, identifying research gaps, and highlighting future directions—ranging from expanding attack goals and tasks to account for new modalities, state-of-the-art models, tools, and pipelines, to underscoring the importance of studying real-world threats in complex scenes.

1 Introduction

Differentiable rendering has emerged as a powerful tool for solving inverse problems in vision and graphics by enabling gradient propagation through the rendering process. Recent methods like Neural Radiance Fields (NeRF) [Mildenhall *et al.*, 2020] and 3D Gaussian Splatting [Kerbl *et al.*, 2023] enable novel view synthesis from limited images to reconstruct 3D models or scenes. These advancements have spurred open-source tools, such as PyTorch3D[1] and user-friendly platforms[2] that allow creating textured 3D models from photos.

Differentiable rendering has also exposed vulnerabilities in DNNs by enabling adversarial attacks. Adversaries exploit DNN gradients to optimize inputs, training, or outputs for malicious purposes, leading to misclassifications in systems such as stop signs in cars, LiDAR [Cao *et al.*, 2019], facial recognition , and 3D models [Xiao *et al.*, 2019]. Similarly, differentiable rendering allows attackers to optimize 3D scene parameters (objects, materials, lighting) via loss gradients. Research on differentiable rendering-based attacks is scattered across:

1. **Attack goals**: e.g., inducing misclassifications or motion/depth errors;
2. **Attackable components**: e.g., preprocessing steps or during inference;
3. **Scene manipulation**: e.g., targeting texture, geometry, or combinations thereof.

In other words, progress in adversarial attacks using differentiable rendering has been made, but systematic comparisons, summaries of strengths, and research gap identification remain challenging. Fig. 1 shows how our survey addresses this gap by organizing tasks like texture manipulation, illumination changes, and 3D mesh alterations, emphasizing both techniques and potential exploitation by adversaries.

1.1 Related Survey and Methodology

This is the first survey to focus on task-based differentiable rendering capabilities for 3D adversarial attacks. Existing work separates differentiable rendering and adversarial research. Kato et al. [2020] briefly mention adversarial attacks as an open problem but do not propose a framework distinguishing attacks using detailed goals and tasks. Since then, NeRF and 3D Gaussian Splatting have gained prominence, requiring discussion. Surveys on NeRF [Xie *et al.*, 2022; Tewari *et al.*, 2022; Gao *et al.*, 2023; Mittal, 2024] and 3D Gaussian Splatting [Chen and Wang., 2024; Tosi *et al.*, 2024] do not address adversarial use. Existing adversarial attack surveys cover 2D/3D models [Li *et al.*, 2024b], robustness and defenses [Miller *et al.*, 2020], or image classification [Machado *et al.*, 2023] but omit differentiable rendering.

We reviewed 28 works from top venues in computer vision, ML, and graphics, covering differentiable rendering methods (e.g., NeRF, 3D Gaussian Splatting) and their use in adversarial attacks. Using a task-based framework, we categorized attacker goals—texture, illumination, and mesh manipulation—to clarify methodologies and vulnerabilities. As a newer field, differentiable rendering research began in 2014, with adversarial applications emerging in 2019.

[1]https://pytorch3d.org
[2]https://poly.cam

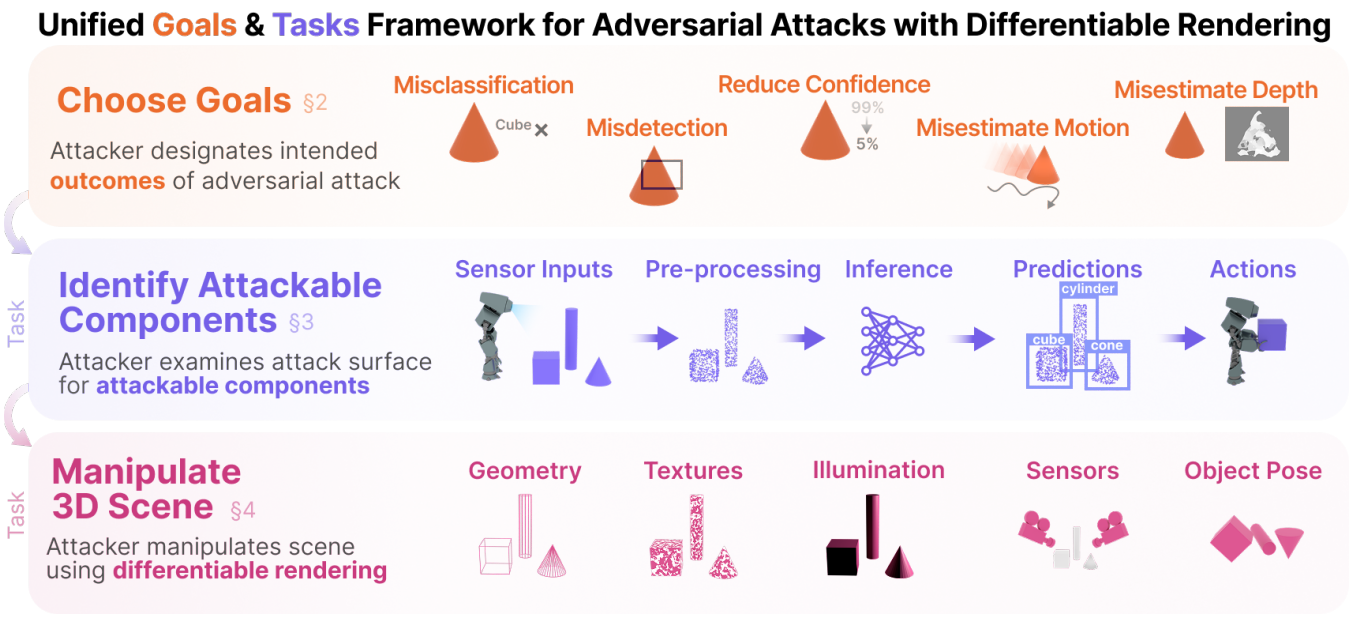

Fig. 1: Visual overview of our unifying survey framework that, by unifying the diverse goals and tasks in identifying attackable components and manipulating scene representations, enables systematic summarization and comparison with existing differentiable rendering related adversarial attack research.

1.2 Contributions

C1. We present the first comprehensive, attacker-task guided survey on adversarial attacks using differentiable rendering, incorporating a use-inspired approach (Fig. 1). Our framework positions each work by attacker objectives and differentiable rendering techniques, defining the attack surface based on feasible scene manipulations of the scene representations (Sec. 3).

- Our methodology links goals to tasks, providing a structured comparison of works and identifying research gaps.
- Table 1 explains differentiable rendering's role in attacks, relevant methods, and current strengths and limitations.

C2. We provide comprehensive categorizations of attack methods, highlighting their impact and real-world implications (Sec. 3). We show a "Target List" of attacked models, including object detection, image classification, and others along with attacker access levels (Table 2). These resources enable researchers to build on existing work, compare outcomes, and develop new techniques to address adversarial threats using differentiable rendering.

C3. We identify key future research directions to address the growing threat of adversarial attacks (Sec. 6). Priorities include developing robust defenses, exploring novel attack strategies, and investigating the physical plausibility of these attacks.

2 Attacker Goals

To better understand how differentiable rendering is used in adversarial attacks, we describe an attacker using the threat model concept to delineate their goals, capabilities, and knowledge in the context of the attack they wish to carry out [Li *et al.*, 2024b]. Using an attacker-task guided perspective, we connect the attacker goals to required tasks and sub-tasks (Sec. 3) that are used in differentiable rendering attacks. Attacker goals encompass any threat affecting the integrity of a DNN's intended task [Papernot *et al.*, 2016b; Li *et al.*, 2024b; Wiyatno *et al.*, 2019]. In this survey, we identify five attacker goals that are used in attacks on deep learning models using differentiable rendering:

G1. Misclassification - the model predicts an incorrect class (untargeted) or a specified incorrect class (targeted) [Papernot *et al.*, 2016a].

G2. Misdetection - manifested as various errors: nothing is detected (evasion), improper bounding box localization, duplicate detections, or detecting background as an object, or combinations thereof [Bolya *et al.*, 2020].

G3. Reduce Confidence - the target class is not predicted with high confidence [Papernot *et al.*, 2016a].

G4. Misestimate Motion - the model misestimates the motion of objects in the scene caused by adversarial movements or objects [Schmalfuss *et al.*, 2023].

G5. Misestimate Depth - the model misestimates depth, affecting the model's ability to perceive distances [Zheng *et al.*, 2024].

In Table 1, we categorized our 28 survey papers as S=Survey (3), M=Metrics (1), or A=Attack (24). Of our 24 Attack papers, we found that 18 works chose goals of inducing misclassifications, 17 induced misdetections, and 10 induced reduction in model confidence while only 1 work each pursued attack goals of misestimation of motion or depth.

	ATTACKER GOALS §2					REQUIRED TASKS §3-4							DOMAIN §5			WHERE
						IDENTIFY §3		MANIPULATE §4								
Work	G.1 Misclassification	G.2 Missed Detection	G.3 Reduce Model Confidence	G.4 Misestimation of Motion	G.5 Misestimation of Depth	A.1 Analyze Attack Surface	A.2 Analyze Scene Components	4.1 Attack Scene Geometry	4.2 Attack Scene Textures	4.3 Attack Scene Object Pose	4.4 Attack Scene Illumination	4.5 Attack Scene Sensors	5.1 Perform Digital Attack	5.2 Perform Physical Attack	Paper Type	Publication Venue
[Abdelfattah *et al.*, 2021]		■						■	■				■		A	ICIP
[Alcorn *et al.*, 2019]	■	■								■			■		A	CVPR
[Bolya *et al.*, 2020]	■	■											■		M	ECCV
[Byun *et al.*, 2022]	■								■				■		A	arXiv
[Cao *et al.*, 2019]	■	■						■					■		A	arXiv
[Dong *et al.*, 2022]	■									■			■	■	A	NeurIPS
[Huang *et al.*, 2024]	■	■						■	■				■	■	A	CVPR
[Li *et al.*, 2024b]	■	■	■			■	■						■		S	ACM CSUR
[Leheng *et al.*, 2023]		■	■						■				■		A	arXiv
[Li *et al.*, 2024a]	■	■							■				■	■	A	arXiv
[Liu *et al.*, 2019a]	■	■						■	■		■		■		A	ICLR
[Machado *et al.*, 2023]	■		■										■		S	ACM
[Maesumi *et al.*, 2021]		■	■						■				■		A	arXiv
[Meloni *et al.*, 2021]	■		■						■				■		A	ICMLA
[Papernot *et al.*, 2016b]	■	■	■			■	■						■		A	EuroS&P
[Papernot *et al.*, 2016a]	■	■	■			■	■						■		A	EuroS&P
[Schmalfuss *et al.*, 2023]				■				■	■	■			■		A	ICCV
[Shahreza and Marcel, 2023]	■					■	■			■		■	■	■	A	TPAMI
[Suryanto *et al.*, 2022]	■	■							■				■		A	CVPR
[Suryanto *et al.*, 2023]	■	■	■						■				■	■	A	ICCV
[Tu *et al.*, 2021]		■	■					■	■				■		A	CoRL
[Wang *et al.*, 2022]	■	■	■						■				■	■	A	AAAI
[Wiyatno *et al.*, 2019]	■	■	■			■	■						■		A	arXiv
[Xiao *et al.*, 2019]	■	■						■					■		A	CVPR
[Yuan *et al.*, 2019]	■	■	■			■	■						■		S	TNNLS
[Zeng *et al.*, 2019]	■									■	■		■	■	A	CVPR
[Zheng *et al.*, 2024]					■				■				■	■	A	CVPR
[Zhou *et al.*, 2024]	■	■	■						■				■	■	A	ICML

Table 1: Overview of representative works on adversarial attacks using differentiable rendering methods. Each row is one work; each column corresponds to a required attacker task or goal. A work's relevant goal or task is indicated by a colored cell. S = Survey, M = Metrics, A = Attack.

3 Identify Attackable Components

To achieve the attacker goals in Sec. 2, one must identify which components can be manipulated by analyzing the attack surface (A1) and scene components (A2).

A1. Analyze Attack Surface. The attack surface includes all data processing stages [Papernot *et al.*, 2016b] in Fig. 1, from sensor inputs and pre-processing to model inference and output actions. In differentiable rendering, this surface extends to the renderer and scene representation, giving adversaries multiple potential entry points. For instance, a robot scanning its 3D environment has:

- **Sensor Inputs** (e.g., camera, LiDAR).
- **Pre-processing** (e.g., generating images or point clouds).
- **Inference** by the DNN model.
- **Predictions** (e.g., labels, bounding boxes, segmentation).
- **Actions** or decisions based on model output.

An attack's effectiveness hinges on the adversary's **access level**: white-box ○, black-box ●, or combination thereof ◐, classified in Table 2. In differentiable rendering, attackers manipulate scene elements, such as object textures or environmental factors (e.g., adversarial weather [Schmalfuss *et al.*, 2023]) to deceive the DNN.

A2. Analyze Scene Components. In differentiable rendering attacks, the 3D *scene representation* is the main target. We categorize its components under: **Geometry** (explicit or implicit [Mildenhall *et al.*, 2020; Kerbl *et al.*, 2023]), **Texture** (color and reflectance), **Position/Pose** (object location/orientation), **Illumination** (light sources, e.g., sun or lamps), and **Sensors** (camera/LiDAR properties like resolution or field of view). Identifying these components helps attackers craft manipulations that produce realistic, adversarial inputs to DNN models.

4 Manipulate 3D Scene

Differentiable rendering enables gradient-based manipulation of any scene elements. With white-box access to victim models, attackers can use loss gradients with respect to scene representations to guide such manipulations. This section reviews common manipulations on 3D scene representations, including geometry, texture, pose, illumination, and sensors. Among the surveyed works, texture attacks are the most prevalent (15) since first adversarial works were on 2D images, followed by geometry (7), pose (5), illumination (2), and sensors (1).

4.1 Attacks on Scene Geometry

Mesh. Attackers use differentiable rendering to generate adversarial meshes by perturbing vertex positions to minimize the cross-entropy loss towards the target label. The adversarial meshes are re-rendered as inputs to victim models. Beyond Pixel Norm-Balls [Liu *et al.*, 2019a] introduced a differentiable rendering framework for generating adversarial geometry V' by propagating gradients through a rendering pipeline via chain rule:

$$V' \leftarrow V - \gamma \frac{\partial C}{\partial I} \frac{\partial I}{\partial N} \frac{\partial N}{\partial V}, \tag{1}$$

where V are vertex positions, N per-face normals, and γ the attack strength. MeshAdv [Xiao *et al.*, 2019] used Neural Mesh Renderer [Kato *et al.*, 2018] to perturb vertices, attacking classifiers and object detectors like YOLO-v3. TT3D [Huang *et al.*, 2024] created adversarial geometry via NeRF and marching cubes but faced scalability challenges due to optimization overhead [Tewari *et al.*, 2022]. Distracting Downpour [Schmalfuss *et al.*, 2023] attacked optical flow models by adding scene-specific spatiotemporally consistent particulate geometry (e.g., rain or snow) to create false motion signals in various datasets [Geiger *et al.*, 2012; Mehl *et al.*, 2023].

Point Cloud. LiDAR-ADV [Cao *et al.*, 2019] used a differentiable LiDAR simulator to perturb point clouds, converting the initially non-differentiable features into differentiable ones with smoothing. Two other works perturbed point cloud objects and converted them to textured meshes to target multi-modal systems [Abdelfattah *et al.*, 2021; Tu *et al.*, 2021].

Geometry Post-Processing and Stabilization. Post-perturbation processing maintains realism and avoids topological issues, such as self-intersections or non-manifold meshes. Techniques like Laplacian smoothing [Vogel and Oman, 1996],

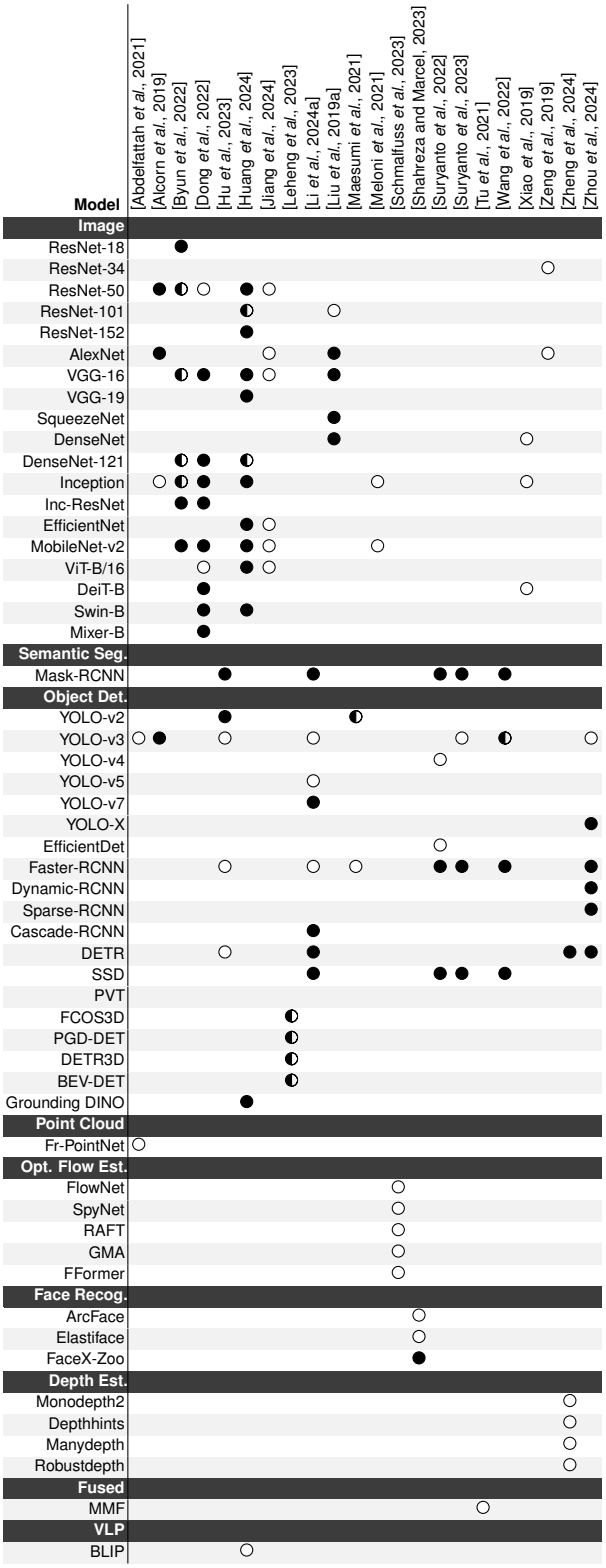

Table 2: DNNs attacked by differentiable rendering using attacker-access levels defined in (A1). Each column is one work; each row is a model.

regularization loss [Liu *et al.*, 2019b], and Chamfer distance loss [Ravi *et al.*, 2020] ensure realistic and stable adversarial geometry. For instance, Laplacian smoothing $\mathcal{L}_S$ minimizes deviations between original vertex $\boldsymbol{v}_i$ and an adversarial vertex $\boldsymbol{v}_q$,

$$\mathcal{L}_S = \sum_{\boldsymbol{v}_i \in V} \sum_{\boldsymbol{v}_q \in \mathcal{N}(\boldsymbol{v}_i)} \|\Delta \boldsymbol{v}_i - \Delta \boldsymbol{v}_q\|_2^2 \tag{2}$$

while Chamfer distance loss $\mathcal{L}_C$ penalizes dissimilarities between pairs of point clouds P and Q.

$$\mathcal{L}_C = \sum_{p \in P} \min_{q \in Q} \|p - q\|_2^2 + \sum_{q \in Q} \min_{p \in P} \|p - q\|_2^2 \tag{3}$$

Depth completion and lighting approximation from Tu et al. [2021] enhance realism by restricting adversary scale within axis-aligned bounding boxes.

4.2 Attack Scene Texture

Texture adversarial attacks manipulate an object's appearance by perturbing its color, pattern, or light reflection properties. Using differentiable rendering, the model's loss gradient is used to perturb the texture mappings (e.g., UV maps) via world-aligned methods that optimize 2D textures, UV map-based methods that directly optimize 3D textures, and neural-rendered methods that dynamically generate textures from 3D representations [Zhou *et al.*, 2024].

Multi-Object Texture Attacks. Two works explore multi-object texture attacks to study transferability. Meloni et al. [2021] create poisoned data by perturbing texels using a saliency map from a non-differentiable renderer. Byun et al. [2022] demonstrate transferability by applying 2D adversarial textures to various 3D objects, achieving successful impersonation and dodging attacks against facial recognition classifiers.

Adversarial Camouflage. Adversarial camouflage targets vehicles and humans. For vehicles, FCA [Wang *et al.*, 2022] applied adversarial textures to an Audi e-Tron in CARLA scenes using the Neural Mesh Renderer (NMR) , while DTA [Suryanto *et al.*, 2022] used EoT for texture projections for a Tesla Model 3 and ACTIVE [Suryanto *et al.*, 2023] made a further improvement with tri-planar mapping, allowing complex shapes. Li et al. [2024a] conducted a flexible physical camouflage attack (FPA) using diffusion models to generate UV-map-based textures, improving the environmental adaptability in neural rendering. RAUCA [Zhou *et al.*, 2024] extended this by incorporating environmental conditions via an encoder-decoder Environmental Feature Extractor (EFE) for optimized textures. For humans, Maesumi et al. [2021] developed adversarial clothing using UV maps and SMPL models, using Blender's subdivision surface modifier to improve texture resolution for more effective attacks.

Texture Attacks on Autonomous Driving Systems. Abdelfattah et al. attacked object textures by treating vertex colors as learnable parameters, reducing YOLOv3 detection in cascaded models used in self-driving [Tu *et al.*, 2021]. Adv3D [Leheng *et al.*, 2023] used NeRF with semantic branch augmentation along with EoT to enhance physical transferability and reduce the confidence of LiDAR detector, while $3D^2Fool$ [Zheng *et al.*, 2024] developed object-agnostic adversarial patches via EoT and texture conversion to attack monocular depth estimation models.

Texture Post-Processing and Stabilization. To enhance the appearance and physical transferability of adversarial textures, many works incorporate post-processing techniques such as hyperparameter tuning, Total Variation (TV) loss, Smooth loss, and Non-Printability Score (NPS). TV loss [Mahendran and Vedaldi, 2015] penalizes differences between adjacent texture pixels $\boldsymbol{x}$, reducing noise and promoting smoothness:

$$TV(\boldsymbol{x}) = \sum_{i,j} \left((\boldsymbol{x}_{i,j+1} - \boldsymbol{x}_{ij})^2 + (\boldsymbol{x}_{i+1,j} - \boldsymbol{x}_{ij})^2 \right)^{\frac{1}{2}}.$$

NPS [Sharif *et al.*, 2016] assesses the physical printability of textures by evaluating pixel proximity to printable RGB triplets $P \subset [0,1]^3$:

$$NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|,$$

where a low score indicates higher printability. These methods ensure adversarial textures are both visually plausible and physically realizable.

4.3 Attack Scene Illumination

Illumination manipulation in differentiable rendering attacks is underexplored due to lack of tools that support controlling light across the whole scene in realistic ways, but two works demonstrate its potential. Liu et al. [2019a] used spherical harmonic lighting [Kautz *et al.*, 2002] for global adjustments, optimizing coefficients via the chain rule (Eq. 1) to preserve realism while attacking DNNs. Zeng et al. [2019] manipulated point lights, creating adversarial lighting to mislead DNNs.

4.4 Attack Scene Sensors

While many attacks test robustness to camera angle changes, Shahreza et al. [2023] directly manipulated camera parameters for attacks. Using NeRF, they optimized camera rotations to find face poses capable of impersonating target identities in facial recognition models. Their method reconstructs 3D faces from 2D facial templates, enabling practical presentation attacks, such as digital screen replay or printed photographs, which can be further extended to create wearable face masks for physical impersonation.

4.5 Attack Scene Object Pose/Translation

DNNs are vulnerable to subtle pose or position changes, with tools like 3DB [Leclerc *et al.*, 2022] available for vulnerability exploration. Using differentiable rendering, attackers can generate precise object poses or translations to induce misclassification in arbitrary settings. Alcorn et al. [2019] demonstrated that Inceptionv3 misclassifies 97% of the pose space for ImageNet objects recognized in their canonical poses, with adversarial poses transferring at high rates to AlexNet, ResNet-50, and YOLOv3. Similarly, Zeng et al. [2019] demonstrated adversarial poses misleading Visual Question Answering models

resulting in wrong scene descriptions. ViewFool [Dong *et al.*, 2022] trained NeRF models on 3D objects from BlenderKit[3], sampling 100 images per model, and demonstrated that ViT-B/16 was more robust to pose attacks than ResNet-50.

5 Digital and Physical Attack Domains

In this section, we focus on the challenges of creating and evaluating attacks in both digital and physical domains (see Table 1) and the tools used for attack.

5.1 Attacks in the Digital Domain

Digital attacks often rely on simulations for controlled testing. However, since differentiable renderers (e.g., Mitsuba, PyTorch3D) often lack simulation features, researchers use non-differentiable tools with simulation features instead. RAUCA [Zhou *et al.*, 2024] and FPA [Li *et al.*, 2024a] used Unreal Engine and CARLA, non-differentiable tools that support data capture, diverse scene setups, lighting conditions and self-driving simulations. Two other works produced adversarial textures and meshes using PyTorch 3D and then evaluated their robustness within scenes rendered by non-differentiable Blender and Unity tools [Zeng *et al.*, 2019; Meloni *et al.*, 2021]. TT3D [Huang *et al.*, 2024] created attacked objects using NeRF and then used Blender and Meshlab for testing cross-render transferability.

5.2 Attacks in the Physical Domain

Implementing real-world adversarial attacks poses several challenges, especially when manufacturing adversarial meshes and textures. Post-processing and mesh stabilization may require advanced techniques like Marching Cubes [Tu *et al.*, 2021] to ensure a watertight, non-degenerate mesh that can be 3D printed. Researchers have also developed flexible "universal" attacks that can be 3D printed once and deployed in multiple scenarios without retraining [Abdelfattah *et al.*, 2021]. When applying adversarial textures, high-resolution printing or color constraints (Sec. 4.2) can enhance feasibility; however, covering large surfaces is costly, prompting the use of localized sticker-mode" approaches [Li *et al.*, 2024a] that only modify a small area (e.g., a vehicle door).

6 Future Directions

To further expose DNN vulnerabilities, we propose four directions for differentiable rendering research:

Target Diversity. Many differentiable rendering attacks focus on targeting cars used for autonomous driving. Meanwhile, use of NeRF and 3D Gaussian Splatting has recently expanded into other real-world applications for robotics and unmanned aerial systems (UAS) but remains largely unconsidered in adversarial ML research. Exploring more diverse targets in these applications would expand Task 4.1 and Task 4.2.

SOTA Models and Other Modalities. Existing differentiable rendering attacks mainly target image classifiers and object detectors, with limited work on optical flow, depth estimation, point cloud classifiers, and multi-modal or multi-task fusion models [Abdelfattah *et al.*, 2021]. Attacks on 3D scene understanding and advanced tasks like tracking or video recognition remain underexplored, despite the growing use of robust models in robotics and AR. Future research could also include newer architectures such as EfficientNet, ViT, and DeiT, which could exhibit different vulnerabilities from older models. Exploiting these emerging vulnerabilities would advance attacker goals **G1.–G5.**

Attacks Considering Real-World Phenomena. Current methods use only basic lighting and camera adjustments, such as varying lighting intensity and position or camera resolution. This overlooks complex environmental factors (*e.g.*, variable light shapes, shadows, color) and camera parameters (lens-warping, field of view, focus distance, and exposure) that create new attack surfaces in drones and other camera-equipped systems. Other physical attacks involving placement of lens covers and rolling shutter exploitation [Sayles *et al.*, 2021] are also understudied. Broadening research on such real-world phenomena would strengthen Tasks 4.3, 4.4, and 5.2.

Tools and Pipelines. While simulators like CARLA are widely used for attack research, differentiable rendering libraries often require specialized knowledge and manual scene configuration. Existing GUIs, such as Blender plugins for Mitsuba[4], help export scenes but still demand significant expertise in 3D modeling. More user-friendly interfaces and integrated pipelines for differentiable renderers would streamline digital attacks (Task 5.1), ultimately facilitating transfer to physical scenarios (Task 5.2).

7 Conclusion

Understanding the evolving capabilities of differentiable rendering is essential for safeguarding deep neural networks. This survey presents a task-guided review of adversarial attacks using differentiable rendering, covering manipulations of 3D objects and scenes that compromise applications like image classification and object detection. By categorizing attacker tasks and linking them to goals, we highlight research gaps such as attacks targeting scene parameters (lighting, camera configurations) and the need for user-friendly resources. Future work should explore novel attack methods and practical physical evaluations, facilitating more resilient DNN defenses in this rapidly advancing area.

References

[Abdelfattah *et al.*, 2021] M. Abdelfattah, K. Yuan, Z. Wang, and R. Ward. Towards Universal Physical Attacks On Cascaded Camera-Lidar 3d Object Detection Models. *ICIP*, 2021.

[Alcorn *et al.*, 2019] M. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W. Ku, and A. Nguyen. Strike (With) a Pose: Neural Networks Are Easily Fooled by Strange Poses of Familiar Objects. *CVPR*, 2019.

[3]https://www.blenderkit.com/

[4]https://github.com/mitsuba-renderer/mitsuba-blender

[Bolya *et al.*, 2020] D. Bolya, S. Foley, J. Hays, and J. Hoffman. TIDE: A General Toolbox for Identifying Object Detection Errors. *ECCV*, 2020.

[Byun *et al.*, 2022] J. Byun, S. Cho, M. Kwon, H. Kim, and C. Kim. Improving the Transferability of Targeted Adversarial Examples through Object-Based Diverse Input. *arXiv*, 2022.

[Cao *et al.*, 2019] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li. Adversarial Objects Against LiDAR-Based Autonomous Driving Systems. *arXiv*, 2019.

[Chen and Wang., 2024] G. Chen and W. Wang. A Survey on 3D Gaussian Splatting. *arXiv*, 2024.

[Dong *et al.*, 2022] Y. Dong, S. Ruan, H. Su, C. Kang, X. Wei, and J. Zhu. ViewFool: evaluating the robustness of visual recognition to adversarial viewpoints. *NeurIPS*, 2022.

[Gao *et al.*, 2023] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li. NeRF: Neural Radiance Field in 3D Vision, A Comprehensive Review. *arXiv*, 2023.

[Geiger *et al.*, 2012] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. *CVPR*, 2012.

[Hu *et al.*, 2023] Z. Hu, W. Chu, X. Zhu, H. Zhang, B. Zhang, and X. Hu. Physically Realizable Natural-Looking Clothing Textures Evade Person Detectors via 3D Modeling. *CVPR*, 2023.

[Huang *et al.*, 2024] Y. Huang, Y. Dong, S. Ruan, X. Yang, H. Su, and X. Wei. Towards Transferable Targeted 3D Adversarial Attack in the Physical World. *CVPR*, 2024.

[Jiang *et al.*, 2024] W. Jiang, H. Zhang, X. Wang, Z. Guo, and H. Wang. NeRFail: Neural Radiance Fields-Based Multiview Adversarial Attack. *AAAI*, 2024.

[Kato *et al.*, 2018] H. Kato, Y. Ushiku, and T. Harada. Neural 3D Mesh Renderer. *CVPR*, 2018.

[Kato *et al.*, 2020] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W.Kehl, and A. Gaidon. Differentiable Rendering: A Survey. *arXiv*, 2020.

[Kautz *et al.*, 2002] J. Kautz, P. Sloan, and J. Snyder. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02. Eurographics Association, 2002.

[Kerbl *et al.*, 2023] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[Leclerc *et al.*, 2022] G. Leclerc, H. Salman, A. Ilyas, S. Vemprala, L. Engstrom, V. Vineet, K. Xiao, P. Zhang, S. Santurkar, G. Yang, A. Kapoor, and A. Madry. 3DB: A Framework for Debugging Computer Vision Models. *NeurIPS*, 2022.

[Leheng *et al.*, 2023] L. Leheng, Q. Lian, and Y. Chen. Adv3D: Generating 3D Adversarial Examples in Driving Scenarios with NeRF. *arXiv*, 2023.

[Li *et al.*, 2024a] Y. Li, W. Tan, C. Zhao, S. Zhou, X. Liang, and Q. Pan. Flexible Physical Camouflage Generation Based on a Differential Approach. *arXiv*, 2024.

[Li *et al.*, 2024b] 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 Computing Surveys*, 56(6), 2024.

[Liu *et al.*, 2019a] H. Liu, M. Tao, C. Li, D. Nowrouzezahrai, and A. Jacobson. Beyond Pixel Norm-Balls: Parametric Adversaries using an Analytically Differentiable Renderer. *ICLR*, 2019.

[Liu *et al.*, 2019b] S. Liu, W. Chen, T. Li, and H. Li. Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning. *ICCV*, 2019.

[Machado *et al.*, 2023] G. Machado, E. Silva, and R. Goldschmidt. Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective. *ACM Computing Surveys*, 55(1), 2023.

[Maesumi *et al.*, 2021] A. Maesumi, M. Zhu, Y. Wang, T. Chen, Z. Wang, and C. Bajaj. Learning Transferable 3D Adversarial Cloaks for Deep Trained Detectors. *arXiv*, 2021.

[Mahendran and Vedaldi, 2015] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. *CVPR*, 2015.

[Mehl *et al.*, 2023] L. Mehl, J. Schmalfuss, A. Jahedi, Y. Nalivayko, and A. Bruhn. Spring: A High-Resolution High-Detail Dataset and Benchmark for Scene Flow, Optical Flow and Stereo. *CVPR*, 2023.

[Meloni *et al.*, 2021] E. Meloni, M. Tiezzi, L. Pasqualini, M. Gori, and S. Melacci. Messing Up 3D Virtual Environments: Transferable Adversarial 3D Objects. *ICMLA*, 2021.

[Mildenhall *et al.*, 2020] B. Mildenhall, P. Srinivasan, M. Tancik, J. Barron, N. Ramamoorthi, and R. Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *ECCV*, 2020.

[Miller *et al.*, 2020] D. Miller, Z. Xiang, and G. Kesidis. Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks. *Proceedings of the IEEE*, 108(3), 2020.

[Mittal, 2024] A. Mittal. Neural Radiance Fields: Past, Present, and Future. *arXiv*, 2024.

[Papernot *et al.*, 2016a] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Celik, and A. Swami. The Limitations of Deep Learning in Adversarial Settings. *EuroS&P*, 2016.

[Papernot *et al.*, 2016b] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. Towards the Science of Security and Privacy in Machine Learning. *EuroS&P*, 2016.

[Ravi *et al.*, 2020] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3D Deep Learning with PyTorch3D, 2020.

[Sayles *et al.*, 2021] A. Sayles, A. Hooda, M. Gupta, R. Chatterjee, and E. Fernandes. Invisible Perturbations: Physical

Adversarial Examples Exploiting the Rolling Shutter Effect. *CVPR*, 2021.

[Schmalfuss *et al.*, 2023] J. Schmalfuss, L. Mehl, and A. Bruhn. Distracting Downpour: Adversarial Weather Attacks for Motion Estimation. *ICCV*, 2023.

[Shahreza and Marcel, 2023] H. Shahreza and S. Marcel. Comprehensive Vulnerability Evaluation of Face Recognition Systems to Template Inversion Attacks via 3D Face Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12), 2023.

[Sharif *et al.*, 2016] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *CCS*, 2016.

[Suryanto *et al.*, 2022] N. Suryanto, Y. Kim, H. Kang, H. Larasati, Y. Yun, T. Le, H. Yang, S. Oh, and H. Kim. DTA: Physical Camouflage Attacks using Differentiable Transformation Network. *CVPR*, 2022.

[Suryanto *et al.*, 2023] N. Suryanto, Y. Kim, H. Larasati, H. Kang, T. Le, Y. Hong, H. Yang, S. Oh, and H. Kim. ACTIVE: Towards Highly Transferable 3D Physical Camouflage for Universal and Robust Vehicle Evasion. *ICCV*, 2023.

[Tewari *et al.*, 2022] A. Tewari, J. Thies, and B. Mildenhall. Advances in Neural Rendering. *Computer Graphics Forum*, 41(22), 2022.

[Tosi *et al.*, 2024] F. Tosi, Y. Zhang, Z. Gong, E. Sandström, S. Mattoccia, M.R. Oswald, and M. Poggi. How NeRFs and 3D Gaussian Splatting are Reshaping SLAM: a Survey. *arXiv*, 2024.

[Tu *et al.*, 2021] J. Tu, H. Li, X. Yan, M. Ren, Y. Chen, M. Liang, E. Bitar, E. Yumer, and R. Urtasun. Exploring Adversarial Robustness of Multi-sensor Perception Systems in Self Driving. *CoRL*, 2021.

[Vogel and Oman, 1996] C. R. Vogel and M. E. Oman. Iterative Methods for Total Variation Denoising. *SIAM Journal on Scientific Computing*, 17(1), 1996.

[Wang *et al.*, 2022] D. Wang, T. Jiang, J. Sun, W. Zhou, X. Zhang, Z. Gong, W. Yao, and X. Chen. FCA: Learning a 3D Full-coverage Vehicle Camouflage for Multi-view Physical Adversarial Attack. *AAAI*, 2022.

[Wiyatno *et al.*, 2019] R. Wiyatno, A. Xu, O. Dia, and A. de Berker. Adversarial Examples in Modern Machine Learning: A Review. *arXiv*, 2019.

[Xiao *et al.*, 2019] C. Xiao, D. Yang, B. Li, J. Deng, and M. Liu. MeshAdv: Adversarial Meshes for Visual Recognition. *CVPR*, 2019.

[Xie *et al.*, 2022] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural Fields in Visual Computing and Beyond. *arXiv*, 2022.

[Yuan *et al.*, 2019] X. Yuan, P. He, Q. Q. Zhu, and X. Li. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9), 2019.

[Zeng *et al.*, 2019] X. Zeng, C. Liu, Y. Wang, W. Qiu, L. Xie, Y. Tai, C. Tang, and A. Yuille. Adversarial Attacks Beyond the Image Space. *CVPR*, 2019.

[Zheng *et al.*, 2024] J. Zheng, C. Lin, J. Sun, Z. Zhao, Q. Li, and C. Shen. Physical 3D Adversarial Attacks against Monocular Depth Estimation in Autonomous Driving. *CVPR*, 2024.

[Zhou *et al.*, 2024] J. Zhou, L. Lyu, D. He, and Y. Li. RAUCA: A Novel Physical Adversarial Attack on Vehicle Detectors via Robust and Accurate Camouflage Generation. *ICML*, 2024.