Towards Effective and Transferable Physical Camouflage against Multi-View BEV-based 3D Perception in Autonomous Driving

Linye Lyu, Jiawei Zhou, Daojing He, Yu Li
Harbin Institute of Technology, Shenzhen
Zhejiang University
{lyulinye, zhoujiawei6666, yu.li.sallylee}@gmail.com
hedaojinghit@163.com

Abstract

Modern autonomous driving systems widely adopt multi-view Bird's-Eye-View (BEV) based 3D perception models due to their superior performance. Despite their success, the robustness of these models against adversarial camouflage attacks remains largely unexplored. Existing camouflage attacks primarily target single-view 2D detectors, limiting their effectiveness in multi-view 3D perception settings. To address this gap, we propose BEVCA, a novel adversarial camouflage generation framework tailored for multi-view BEV-based 3D perception. Our framework integrates a BEV-feature-based adversarial loss with a multi-view neural rendering module, enabling effective and transferable camouflage attacks across different models and tasks. Extensive experiments in both digital and physical settings demonstrate that BEVCA outperforms state-of-the-art baselines and exhibits strong robustness under real-world conditions. Our code is available at https://github.com/WAVE-AT-IDEA/BEVCA.

1 Introduction

Vision-based autonomous driving (AD) systems have been widely adopted by major vehicle manufacturers, such as Tesla [17], due to their competitive performance. These systems utilize multiple camera sensors surrounding the ego vehicle to perceive the nearby environment with deep learning models [10; 23; 21; 4; 20; 7]. However, despite the significant progress, current research shows that deep neural networks (DNNs) are vulnerable to adversarial examples [16], which threaten the safety-critical nature of AD systems.

Prior attacks mainly focus on the single-view 2D vehicle detection task. These methods [18; 14; 15; 24; 9; 25] use the Carla simulator [2] to generate vehicle images with various transformations to enhance the attack robustness. Besides, they leverage a differentiable neural renderer [5; 11] to enable direct adversarial camouflage optimization against the target model.

Despite impressive progress in 2D attacks, current state-of-the-art vehicle detection methods in autonomous driving are dominated by multi-view 3D perception models. As a result, the current single-view 2D attack pipelines cannot be directly applied to the multi-view 3D perception models. Some works, such as [26; 19], explore attacking multi-view 3D perception models with adversarial patches by minimizing detection scores. However, the adversarial patches are inherently not robust against various viewing angles since they cannot fully cover the target vehicle, leading to suboptimal attack performance. Besides, attacks in the real world are mostly black-box settings. Yet, all previous methods rely on the target model's detection scores to optimize the adversarial texture, thereby limiting their attack transferability.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

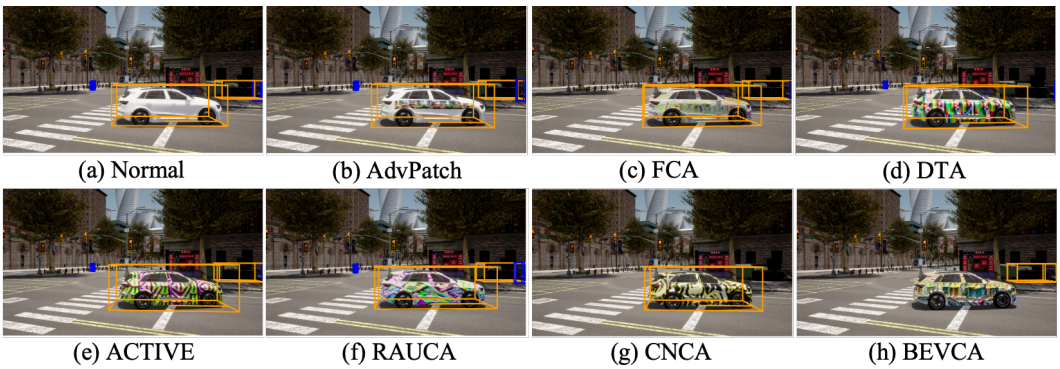


Figure 1: Comparison of 3D detection results using different textures on the photo-realistic simulation environment, where only the adversarial camouflage generated by our BEVCA framework successfully evades detection. (a) A vehicle with normal painting; (b) AdvPatch [19]. (c)-(g) are the top-performed methods against 2D detection: FCA [18], DTA [14], ACTIVE [15], RAUCA [24], and CNCA [9] respectively; (h) Our method BEVCA.

To tackle the above challenges, we propose BEVCA, a novel framework for generating adversarial camouflage against multi-view BEV-based 3D perception models. Our key observation is that most state-of-the-art multi-view 3D perception models [10; 4; 20; 7] rely on Bird's-Eye-View (BEV) features, a unified 2D spatial representation that supports various downstream tasks in AD systems [3; 22]. This insight suggests that attacking BEV features could yield transferable adversarial effects across different models and tasks. However, we find that directly optimizing camouflage using naive BEV features is ineffective (see Section 3). The main challenge is to identify the BEV feature regions that are highly sensitive to the target vehicle's texture . To address this, we propose a dedicated masking technique that accurately locates the BEV feature areas highly relevant to the target car and design adversarial loss accordingly. In addition, we extend the current single-view neural rendering to multi-view neural rendering to generate realistic, consistent adversarial vehicle images across multiple camera views, enabling gradient-based texture optimization. The combination of the BEV-feature-based attack and multi-view neural rendering enables highly effective, transferable adversarial camouflage across diverse BEV-based perception tasks and models.

Our main contributions are summarized as follows:

- We present a BEV-feature-based Camouflage Attack (BEVCA), a framework for generating adversarial camouflage against multi-view BEV-based 3D perception models. It achieves effective and transferable adversarial camouflage attacks using BEV-feature-level attacks and multi-view neural rendering.
- We propose a novel mask-enhanced BEV-based adversarial loss that accurately targets car-texture-sensitive regions of the BEV, enabling strong transferability across diverse BEV-based 3D perception models and downstream tasks.
- We extend single-view neural rendering to multi-view neural rendering, producing realistic, consistent vehicle images accross multiple viewpoints relative to the ego vehicle, facilitating efficient gradient-based adversarial camouflage optimization.

We conduct a comprehensive evaluation with various multi-view BEV-based 3D perception models. The results show that our adversarial camouflage outperforms existing baselines in black-box settings, achieving average improvements of 36.2% and 21.6% in 3D object detection and segmentation, respectively. Furthermore, we also validate the robustness of our approach in the physical world, achieving an average attack success rate of 51.18% in black-box settings.

2 Related work and threat model

2.1 Related work

In this section, we first introduce multi-view 3D perception, then review adversarial camouflage attacks on vehicle detection, and finally define the threat model for our scenario.

Multi-view 3D Perception. The development of large-scale multi-view datasets such as NuScenes [1] and Waymo [13] has significantly accelerated progress in multi-view 3D perception. Bird's-Eye-View (BEV) models [10; 4; 20; 7] have demonstrated state-of-the-art performance across various autonomous driving (AD) tasks. These models extract and aggregate information from multi-view driving images to construct a unified BEV feature representation of the environment, which can support various downstream tasks, including 3D object detection, map segmentation, and even end-to-end driving systems [3; 22].

Adversarial Camouflage Accurate vehicle detection is critical for the safety of AD systems, motivating a growing interest in studying the robustness of vehicle detectors with adversarial attacks. These attacks can be categorized as single-view 2D attacks and multi-view 3D attacks. Existing 2D attack methods leverage 3D simulators such as Carla [2] to generate target vehicle images under various transformations for optimizing adversarial textures. Then, these methods [18; 14; 15; 24; 9] employ the differentiable neural renderer [5; 11] to directly optimize adversarial textures via gradient backpropagation. Besides, [14; 15; 24; 9] further incorporate realistic environmental features, such as shadows and fog, to mitigate the domain gap between simulation and real-world conditions, achieving better attack performance. Regarding multi-view 3D attacks, previous works attempt to generate the adversarial patches on the target vehicle. [26; 19] utilize the 3D annotations of the target vehicle to transform the 2D adversarial patches so that they look realistic and consistent in the multi-view image setting, while [6] utilizes NeRF to generate adversarial texture on the vehicle.

In contrast to prior works, our framework introduces a novel combination of multi-view neural rendering and BEV-feature-based adversarial attack to enable effective and transferable attacks on multi-view BEV-based 3D perception models.

2.2 Threat Model

Targeted Model: We consider attackers targeting multi-view perception models that construct a BEV feature representation for downstream tasks like object detection and segmentation. These BEV perception models, for instance, BEVFormer [7], are widely adopted in the current state-of-the-art multi-view 3D perception tasks in autonomous driving [3; 22].

Attacker: The attacker's goal is to reduce the perception performance of the target models towards the camouflaged vehicle, subject to physical constraints and consistently effective across various viewpoints. This is achieved by altering the vehicle's appearance with adversarial camouflage textures. The attack operates in two distinct phases: firstly, ***offline optimization*** where attacker leverages the simulation environment (e.g., CARLA) to obtain ground-truth target vehicle positions and generate perfectly paired no-vehicle reference scenes, and performs gradient-based optimization against the victim model (white-box). This information is strictly confined to the offline preparation phase. Secondly, ***passive deployment***, where the adversarial texture is printed and physically applied to the target vehicle. In the black-box setting, the attacker additionally assumes only that the victim model relies on BEV feature representation, with no internal access to model weights or intermediate features. In both settings, the attacker cannot modify training data, model weights, or sensor configurations.

3 Methods

In this section, we first present an overview of our framework for generating effective and transferable adversarial camouflage against multi-view BEV-based 3D perception models. Then, we describe the details of the essential components.

3.1 Overview

Figure 2 illustrates the overall framework for generating adversarial camouflage. We first construct a multi-view image dataset of a target vehicle captured from the ego vehicle's perspective using the

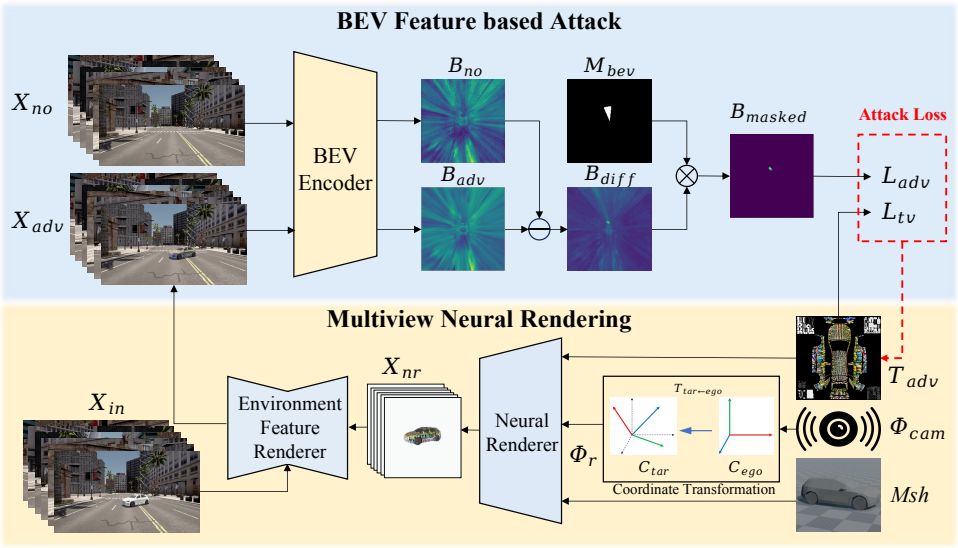


Figure 2: The overview of BEVCA. Our framework consists of multi-view neural rendering and BEV-feature-based attack modules. The multi-view neural rendering outputs the multi-view camouflage vehicle images from the ego vehicle perspective. The BEV-feature-based attack locates the BEV feature regions that are highly associated with the target vehicle with a designed mask. The combination of both modules enables effective and transferable camouflage attacks against multi-view 3D perception models.

Carla simulator [2], following the NuScenes [1] dataset format. This dataset contains the multi-view images X_{in}, camera configurations Φ_{cam}, and the transformation matrices $T_{tar\leftarrow ego}$ that map ego coordinates to target vehicle coordinates. Our multi-view neural renderer first uses a neural renderer NR, which takes the adversarial texture T_{adv}, 3D vehicle mesh Msh, camera configurations Φ_{cam}, and the transformation matrices $T_{tar\leftarrow ego}$ as input, and produces the multi-view rendered adversarial target vehicle images X_{nr}. Subsequently, both X_{nr} and X_{in} are input into an Environmental Feature Renderer (EFR). We use the pretrained EFR from [9] to extract environmental features from X_{in} and render them together with the background to output the realistic multi-view camouflaged target vehicle images X_{adv}. To summarize:

$$X_{adv} = EFR(X_{in}, NR(Msh, T_{adv}, \Phi_{cam}, T_{tar\leftarrow ego})) \tag{1}$$

Instead of attacking task-specific output scores (e.g., detection confidence), we target the BEV feature representation to improve attack transferability. A key challenge is identifying the regions within the BEV feature map that correspond to the target vehicle's presence in the multi-view images. To this end, we construct a *no-vehicle* multi-view dataset X_{no} under the identical settings as X_{in}, except that the target vehicle is removed. We feed X_{adv} and X_{no} into a BEV encoder to obtain the corresponding BEV features:

$$B_{adv} = BEV(X_{adv}) \tag{2}$$
$$B_{no} = BEV(X_{no}) \tag{3}$$

Since the only difference between X_{adv} and X_{no} is the presence of the target vehicle, we hypothesize that the difference between B_{adv} and B_{no} reflects the target vehicle's impact on BEV feature space:

$$B_{diff} = |B_{adv} - B_{no}| \tag{4}$$

Empirically, we observe that the prominent regions in B_{adv} are geometrically aligned with the target vehicle's ground-truth location. Based on this observation, we apply a BEV mask M_{bev} to B_{diff} , locating the affected regions B_{masked} :

$$B_{masked} = B_{diff} \cdot M_{bev} \tag{5}$$

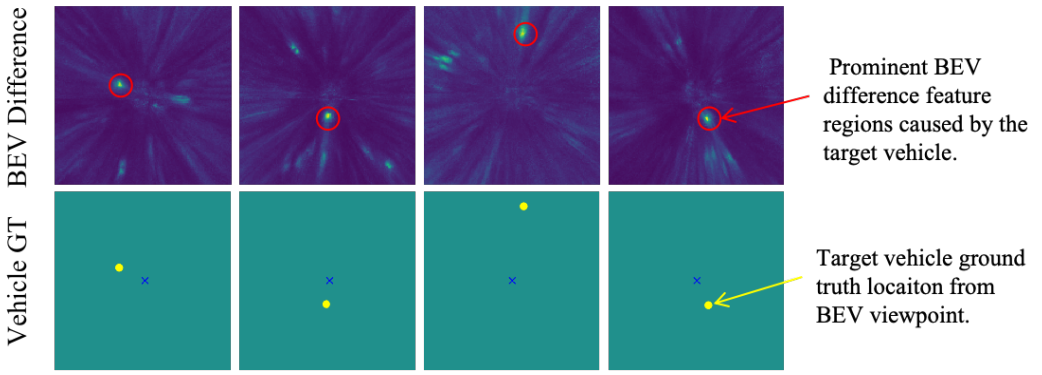


Figure 3: Examples of BEV difference feature and corresponding vehicle ground truth images. We can observe a strong correlation between BEV regions relevant to the target vehicle (marked by the red circles) and vehicle ground truth position (marked by the yellow points).

We define an adversarial loss L_{adv} based on B_{masked} to attack the BEV representation towards the target vehicle. Additionally, following [12], we employ a total variation loss L_{tv} on the adversarial texture image T_{adv} to ensure the naturalness of texture to human vision. The details of L_{adv} and L_{tv} will be discussed in the following section. Finally, our total loss function to optimize the adversarial texture is:

$$L_{total} = \alpha L_{adv} + \beta L_{tv}, \tag{6}$$

where α, β are the hyperparameters to control the contribution of each loss function.

3.2 Multi-view Neural Rendering

Previous works [14; 15; 24; 9] have demonstrated the effectiveness of adversarial camouflage against 2D object detectors by leveraging realistic neural rendering of textured vehicles. In the single-view setting, the adversarial vehicle is placed at the center of the image, and the rendering can be achieved using the object-centric rendering methods from [11; 5]. These methods only require the relative camera pose with respect to the adversarial vehicle, which can be easily obtained during the camera initialization in CARLA. However, this pipeline is not directly applicable to multi-view 3D perception tasks. In such scenarios, multiple cameras are attached to the ego vehicle to observe the target vehicle. As a result, the camera pose relative to the target vehicle—required by the renderer—can no longer be retrieved through the simple initialization procedure used in the single-view pipeline. To support multi-view rendering, it is necessary to redesign the pipeline by computing the relative pose between each ego-mounted camera and the target vehicle, based on the geometric relationships among the ego vehicle, the target vehicle, and the cameras.

To address this, we extend the popular differentiable rendering library PyTorch3D [11] as the naive renderer to support multi-view rendering. The naive renderer requires three key inputs: the 3D mesh of the vehicle, the adversarial texture, and the camera settings relative to the target vehicle. The key challenge is to compute the transformations $T_{tar \leftarrow ego}$, which convert the camera settings of each camera coordinate system Φ_{cam} to the target vehicle's coordinate system:

$$\Phi_r = T_{tar \leftarrow ego} \cdot \Phi_{cam} \tag{7}$$

where Φ_r can be directly input to the naive neural renderer. These transformations ensure that each rendered view accurately reflects the camera's perspective towards the target vehicle. We apply these camera parameter transformations to all six ego-mounted cameras and perform image rendering as defined in Eq 1 to obtain multi-view rendering images. The full mathematical details and transformation steps are provided in the appendix A for interested readers.

3.3 BEV-Feature-based Adversarial Attack

The current state-of-the-art multi-view 3D perception models [10; 4; 20; 7] rely heavily on BEV features to support downstream perception tasks in autonomous driving systems [3; 22]. This

motivated us to attack from the BEV feature to achieve an effective and transferable attack. However, since the BEV feature is high-dimensional, it is challenging to directly isolate the feature areas that are influenced by the target vehicle. To tackle this, we utilize the BEV feature extracted from the dataset with the same camera configurations but excluding the target vehicle. We analyze the difference between these two BEV features to locate the target vehicle's associated feature regions, which we used to calculate the adversarial loss.

Specifically, as shown in Eq. 2 and 3, we first obtain the BEV features B_{adv} and B_{no} from X_{adv} and X_{no} respectively. We compute their difference to obtain the B_{diff}, which has the shape of (H, W, C), where H and W define the BEV perception grid, and C is the feature dimension. To localize the affected regions, we average across the feature dimension C and visualize the resulting 2D heatmap. Empirically, as shown in Figure 3, we observe that high-activation areas in this map tend to coincide with the ground-truth location of the target vehicle, suggesting a strong correlation.

Based on the above observation, we propose a BEV mask $\mathcal{M}_{bev}$ to effectively crop out the affected regions. We leverage the annotated position of the target vehicle to build this mask. For convenience, we represent the target's position in polar coordinates (r_{tar}, θ_{tar}) related to the ego vehicle and define the mask region as:

$$\mathcal{M}_{bev} = \{(r, \theta) \,|\, |r - r_{\text{tar}}| \leq \Delta r,\ |\theta - \theta_{\text{tar}}| \leq \Delta\theta\} \tag{8}$$

where Δr and $\Delta\theta$ are the tunable thresholds to control the mask area. In the ablation section, we evaluate the effectiveness of this mask strategy.

Once the masked BEV difference B_{masked} is obtained (as shown in Eq. 5.), we define the adversarial L_{adv} as the mean of the non-zero elements in B_{masked}:

$$L_{adv} = \frac{\sum_i B_{masked,i}}{\sum_i \mathbb{1}(B_{masked,i} \neq 0)} \tag{9}$$

where $\mathbb{1}(\cdot)$ is the indicator function, which returns 1 if the condition inside is true and 0 otherwise. The denominator represents the number of non-zero elements in B_{masked}, and the numerator is the sum of all elements in B_{masked}.

Additionally, we also apply the Total Variation (TV) loss followed by [12] to encourage naturalness of the texture towards human vision:

$$L_{tv} = \sum_{i,j} \left(x_{i,j} - x_{i+1,j}\right)^2 + \left(x_{i,j} - x_{i,j+1}\right)^2 \tag{10}$$

4 Experiments

4.1 Experimental Settings

Datasets. We utilize the Carla simulator [2] to generate datasets for our experiments. To have a comparative analysis with prior studies [18; 15], we select the Audi E-Tron as the target vehicle model. We generate a dataset of 2500 samples for camouflage generation and a test dataset of 500 samples for camouflage evaluation. These datasets contain multi-view images of the target vehicle with various distances and view angles relative to the ego vehicle, across various background scenes. During our experiments, we found that the performance of the original targeted models degraded significantly in the simulation dataset compared to their reported results, as shown in Table 7. This is because these models are trained on realistic datasets like nuScenes [1]. To alleviate the domain gap between simulation and realism, we generate a fine-tuning dataset of 500 samples with completely different settings from camouflage generation and test datasets to adapt these models to the digital domain. For the real-world evaluation, we print our camouflage in a 2D UV map image and stick it to a full-size Audi vehicle. We build a simple multi-view camera data collection platform to collect real-life multi-view data. We collect three physical datasets: one of the car with normal painting, one of the car with CNCA texture, and one of the car with our camouflage. Each dataset has 150 samples with different azimuths and distances towards the ego position.

Baselines. We compare BEVCA with five adversarial camouflage methods targeting 2D vehicle detection—FCA [18], DTA [14], ACTIVE [15], RAUCA [24], and CNCA [9]—and AdvPatch

[19], an adversarial patch method targeting 3D detection. For each baseline, we apply its officially released adversarial texture to the Audi E-Tron model and evaluate it using the same multi-view BEV perception pipeline. These baseline textures are not re-optimized or adapted to our experimental setting.

Evaluation metrics. In digital experiment, to evaluate the effectiveness of the adversarial camouflage, we utilize the Average Precision (AP) score defined by the 2D center distance between the prediction and ground truth on the ground plane [1; 7] for 3D detection task and Intersection-over-Union (IoU) [23] of vehicle class for BEV segmentation task, respectively. In the real-world experiments, we report Attack Success Rate (ASR), defined as the percentage of target vehicles detected in the clean car painting that are no longer detected in the adversarial camouflage. Detection is determined using an IoU threshold of 0.5, with the same evaluation criteria applied to clean and adversarial cases.

Target models. We choose the BEVFormer-base [7] model as the white-box target model for adversarial camouflage generation. To evaluate the effectiveness and transferability of the optimized camouflage, we utilize a collection of 3D object detection and BEV segmentation models treated as black-box models. 3D object detection models include BEVFormer series (small, tiny, base-seg-det, small-seg-det) [7] with different model configurations. The BEV segmentation models include Lift-Splat-Shoot [10], Cross View Transformer [23] and SinBEVT [21]. We train the original version of these models on the fine-tuning dataset with one epoch before evaluating on the test dataset.

Optimization details. We utilize the Adam optimizer with a learning rate of 0.01 for adversarial camouflage generation. The camouflage texture is initialized randomly and trained with 10 epochs. For the hyperparameters during camouflage optimization, we set the values of α and β (see Eq. 6) to 1 and 1000 respectively, as shown in Figure 8. For the BEV mask, we set the distance and angle thresholds $(\Delta r, \Delta\theta)$ to 15 meters and 15 degrees (see Eq. 8) after hyperparameter tunings. We conduct experiments on one NVIDIA RTX A800 80GB GPU with around 20 hours of camouflage optimization time.

4.2 Attack Performance Evaluation

Digital attack on 3D perception models. We compare our proposed method, BEVCA, against several state-of-the-art adversarial camouflage approaches, including FCA, DTA, ACTIVE, RAUCA, CNCA, and AdvPatch. To thoroughly evaluate the effectiveness and transferability of our method, we conduct extensive experiments across a variety of multi-view 3D perception models in both 3D object detection and BEV segmentation tasks. The results are shown in Table 1, where BEVCA consistently outperforms all baselines. AdvPatch shows performance comparable to that of a normally painted vehicle, primarily due to its reliance on small, localized patches. FCA provides only marginal improvement, as it lacks the ability to model complex environmental interactions. In contrast, DTA, ACTIVE, RAUCA, and CNCA offer stronger performance, yet remain suboptimal because they are not explicitly designed for multi-view perception models. Our method surpasses all the baselines, achieving an average performance gain of approximately 36.2% in black-box settings against the best previous baseline, demonstrating strong effectiveness and transferability.

We also provide some adversarial camouflage vehicle examples against 3D detection in different scenarios in the appendix section . These examples demonstrate that the vehicle with normal painting is accurately detected with well-aligned 3D bounding boxes. However, with our adversarial camouflage texture, the target vehicle becomes significantly harder to detect, often resulting in inaccurate bounding boxes or complete disappearance from the detection output.

For the BEV segmentation task, although the adversarial camouflage generated by BEVCA is specifically optimized for the BEVFormer base model, it consistently achieves the best attack performance across all segmentation models when compared to existing baselines. Notably, BEVCA attains an average improvement of approximately 21.6% over the strongest previous baseline performance in black-box settings. These results further underscore the task transferability and robust generalization of our approach across diverse BEV-based perception tasks.

Physical world evaluation. We conduct the physical world evaluation by printing the camouflage in a 2D UV map format and sticking it to a full-size Audi car. To collect real multi-view data, we built a multi-view camera data collection platform following the camera setup from the nuScenes [1] dataset. We collect three datasets: one of the car with normal painting, one of the car with CNCA texture (We choose CNCA because it achieve the best attack among all the baselines in the simulation

Table 1: Comparison of the effectiveness of attacks across various 3D detection and BEV segmentation models. The metrics for 3D detection and BEV segmentation are AP and IoU of the vehicle class, respectively.

METHODS	3D DETECTION					BEV SEGMENTATION		
	BASE	SMALL	TINY	BASE-SEG-DET	SMALL-SEG-DET	LSS	CVT	SINBEVT
NORMAL	0.618	0.721	0.514	0.750	0.634	0.509	0.609	0.570
FCA	0.593	0.696	0.512	0.721	0.595	0.442	0.505	0.489
DTA	0.578	0.663	0.469	0.684	0.591	0.372	0.480	0.449
ACTIVE	0.559	0.664	0.433	0.682	0.588	0.367	0.481	0.436
RAUCA	0.562	0.682	0.479	0.689	0.576	0.380	0.459	0.427
CNCA	0.567	0.626	0.404	0.634	0.523	0.306	0.401	0.360
ADVPATCH	0.601	0.702	0.516	0.703	0.608	0.479	0.579	0.533
BEVCA	**0.191**	**0.285**	**0.381**	**0.329**	**0.331**	**0.253**	**0.307**	**0.273**

Table 2: Attack success rates (ASR %) by methods and models in the physical world evaluation.

METHODS	BEVFORMER BASE	BEVFORMER SMALL	BEVDET-R50	BEVDET-R50-CBGS
CNCA	30.00%	29.49%	42.80%	34.99%
BEVCA	49.60%	46.99%	64.41%	42.13%

results in Table 1.), and the one of the car with our BEVCA camouflage. We evaluate attack success rates across four victim models: BEVFormer-base, BEVFormer-small, BEVDet-R50, and BEVDet-R50-CBGS. BEVFormer [7] and BEVDet [4] are representatives for backward and feedforward BEV-based perception models, respectively [8]. As shown in Table 2, BEVCA achieves substantially higher ASR than CNCA across all victim models with average of 51.18% ASR in black-box settings, demonstrating our method can pose a severe safety threat in real-world applications. Furthermore, Figure 4 shows that our camouflaged car can be successfully undetected from the 3D detector in different camera viewpoints while the other surrounding objects (cars, scooters, pedestrians) are well detected. In summary, the physical evaluation results demonstrate that our method is transferable to the real world.

4.3 Ablation Studies

Effectiveness of different adversarial losses. To validate the necessity of both (i) BEV-feature-level optimization and (ii) our masking scheme, we compare our approach ($L_{bev} + \mathcal{M}_{bev}$) with several alternative adversarial losses against the same target model (BEVFormer-base): L_{det} based on 3D detection scores, L_{enc} based on image-encoded features within the BEV encoder, L_{head} based on hidden-state features from the 3D detection head, L_{bev} based on the whole BEV features without applying masks.

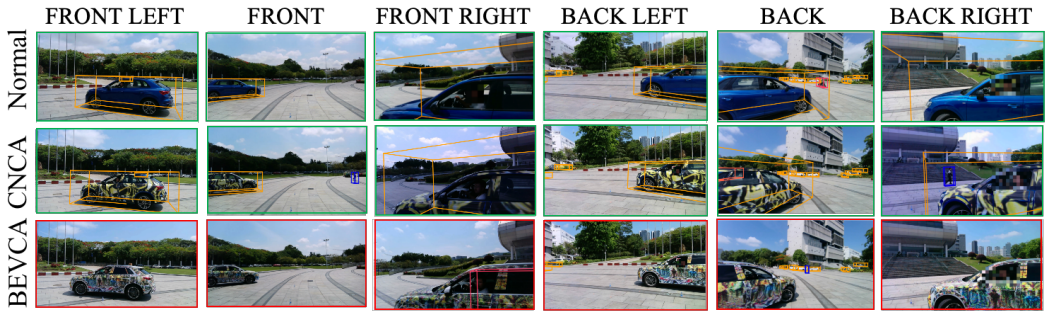


Figure 4: Visualization of our camouflage successful attack examples against 3D object detection in the physical world from different camera viewpoints.

Table 3: The attack performance of different adversarial loss functions across various 3D detection and BEV segmentation models.

METHODS	3D DETECTION					BEV SEGMENTATION		
	BASE	SMALL	TINY	BASE-SEG-DET	SMALL-SEG-DET	LSS	CVT	SINBEVT
L_{det}	0.564	0.683	0.498	0.728	0.597	0.450	0.505	0.467
L_{enc}	0.392	0.564	0.428	0.528	0.529	0.380	0.371	0.333
L_{head}	0.471	0.603	0.485	0.601	0.529	0.367	0.426	0.347
L_{bev}	0.612	0.704	0.508	0.735	0.585	0.498	0.588	0.574
$L_{bev} + \mathcal{M}_{bev}$	**0.191**	**0.285**	**0.381**	**0.329**	**0.329**	**0.253**	**0.307**	**0.258**

Table 3 presents the attack performance across different models in both 3D object detection and BEV segmentation tasks. The results reveal several critical insights: first, directly attacking 3D detection outputs (L_{det}) shows limited effectiveness and transferability, achieving similar attack performance to baselines like FCA; second, attacking non-BEV intermediate features (L_{enc} and L_{head}) yields substantially weaker attack performance compared to our method, indicating suboptimal attacks; most importantly, attacking BEV features without masking (L_{bev}) shows almost no attack effect, demonstrating that naive BEV-level attacks fail due to overwhelming noise from car-irrelevant regions. Since the target vehicle occupies only approximately 3% of the entire BEV perception area (as shown in Figure 3), gradients are dominated by noise from irrelevant regions when the entire BEV feature is used for loss computation. In contrast, our proposed method ($L_{bev} + \mathcal{M}_{bev}$) achieves the strongest attack performance, which demonstrates superior transferability across all black-box models in both 3D detection and BEV segmentation tasks. These results empirically validate that: (1) attacking BEV features achieves both effective and transferable attack performance, and (2) our masking technique is essential for locating the car-texture-sensitive BEV regions, enabling effective adversarial optimization.

In addition, we conduct further ablation analyses in the appendix to provide a more complete understanding of BEVCA. Specifically, we study sensitivity to key optimization hyperparameters (the TV weight β and BEV-mask parameters $\Delta r, \Delta\theta$) in Figure 8, and evaluate robustness under simple defense transformations in Table 6. These additional studies consistently support the conclusions of our main ablation.

5 Conclusion

We have proposed BEVCA, a novel adversarial camouflage attack framework targeting multi-view BEV-based 3D perception. Our framework can generate effective and transferable adversarial camouflage across various multi-view BEV-based 3D perception models and tasks. Our approach consists of a novel BEV-feature-based adversarial loss that enables transferable attacks, and a multi-view neural rendering module that facilitates effective gradient-based camouflage optimization against multi-view perception models. Extensive experiments on both 3D object detection and BEV segmentation tasks demonstrate that BEVCA consistently outperforms existing baselines against diverse black-box models under multi-view perception settings. Moreover, we validate the practical effectiveness of our approach through physical experiments on a full-size vehicle, highlighting its robustness in the real world.

6 Limitations and Future Work

While our physical experiments validate BEVCA's real-world effectiveness, several challenges still remain. First, physical deployment introduces a sim-to-real domain gap caused by factors like UV map deformation, and lighting conditions. Systematically quantifying these variables is left for future work. Second, adversarial textures are inherently bound to a specific vehicle's 3D geometry; transferring the attack to a different vehicle class requires a one-time offline re-optimization. Finally, as autonomous driving evolves toward Vision-Language-Action (VLA) frameworks and World Models, BEV representations remain their one of their foundational spatial features. While our framework exposes immediate risks in BEV perception, investigating the robustness of VLA models against adversarial camouflage is a critical direction for future research.

7 Impact Statement

This work is intended to advance the safety and robustness of autonomous driving (AD) systems. While the proposed adversarial attack method could potentially be misused by malicious users, it primarily serves as a valuable tool for strengthening AD systems through adversarial training (see Appendix Section D), adversarial testing, and adversarial detection. By exposing critical vulnerabilities under realistic physical conditions, our approach ultimately contributes to improving the security and reliability of AD systems.

Acknowledgment

This research is supported by the National Key R&D Program of China (Grant Nos. 2026ZD1500404 and 2024YFE0215300), the National Natural Science Foundation of China (Grant Nos. 62376074 and 62674206), the Guangdong S&T Program (Grant No. 2026B0101100003), and the Shenzhen Science and Technology Program (Grant Nos. KJZD20231023095959002, KCXFZ20230731093001002, KJZD20240903100505007, ZDCY20250901104705006, and KJZD20240903102901003). Yu Li and Daojing He are the corresponding authors of this article.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

[3] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862.

[4] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

[5] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916.

[6] Leheng Li, Qing Lian, and Ying-Cong Chen. 2024. Adv3d: Generating 3d adversarial examples for 3d object detection in driving scenarios with nerf. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10813–10820.

[7] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. 2022. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, page 1–18, Berlin, Heidelberg. Springer-Verlag.

[8] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar, Tong Lu, and Jose M Alvarez. 2023. Fb-bev: Bev representation from forward-backward view transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6919–6928.

[9] Linye Lyu, Jiawei Zhou, Daojing He, and YU LI. 2024. CNCA: Toward customizable and natural generation of adversarial camouflage for vehicle detectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[10] Jonah Philion and Sanja Fidler. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pages 194–210. Springer.

[11] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*.

[12] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1528–1540. ACM.

[13] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[14] Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. 2022. DTA: Physical Camouflage Attacks Using Differentiable Transformation Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15305–15314.

[15] Naufal Suryanto, Yongsu Kim, Harashta Tatimma Larasati, Hyoeun Kang, Thi-Thu-Huong Le, Yoonyoung Hong, Hunmin Yang, Se-Yoon Oh, and Howon Kim. 2023. ACTIVE: Towards Highly Transferable 3D Physical Camouflage for Universal and Robust Vehicle Evasion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4305–4314.

[16] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[17] Tesla, Inc. 2025. Full Self-Driving. `https://www.tesla.com/fsd`. Accessed: 2025-07-30.

[18] Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. 2022. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2414–2422.

[19] Jian Wang, Fan Li, and Lijun He. 2025. A unified framework for adversarial patch attacks against visual 3d object detection in autonomous driving. *IEEE Transactions on Circuits and Systems for Video Technology*.

[20] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on robot learning*, pages 180–191. PMLR.

[21] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. 2022. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning (CoRL)*.

[22] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. 2024. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer.

[23] Brady Zhou and Philipp Krähenbühl. 2022. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13760–13769.

[24] Jiawei Zhou, Linye Lyu, Daojing He, and Yu Li. 2024. Rauca: a novel physical adversarial attack on vehicle detectors via robust and accurate camouflage generation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

[25] Jiawei Zhou, Linye Lyu, Daojing He, and Yu Li. 2024. Toward robust and accurate adversarial camouflage generation against vehicle detectors. *arXiv preprint arXiv:2411.10029*.

[26] Zijian Zhu, Yichi Zhang, Hai Chen, Yinpeng Dong, Shu Zhao, Wenbo Ding, Jiachen Zhong, and Shibao Zheng. 2023. Understanding the robustness of 3d object detection with bird's-eye-view representations in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21600–21610.

A Multi-View Camera Pose Transformation Details

The key challenge for precise multi-view rendering of the target vehicle is to compute the transformation from each camera coordinate system to the target vehicle's coordinate system. First, we can obtain the following geometric data from the CARLA API:

- The global position and rotation of the ego vehicle: t_{ego}^{glob}, R_{ego}^{glob}.
- The global position and rotation of the target vehicle: t_{tar}^{glob}, R_{tar}^{glob}.
- The position and rotation of the ego camera in the ego vehicle frame: t_{cam}^{ego}, R_{cam}^{ego}.

Since CARLA employs a left-handed coordinate system while PyTorch3D uses a right-handed one, we first convert all relevant data accordingly through a proper handedness transformation.

$$\hat{t} = M \cdot t,\ \hat{R} = M \cdot R \cdot M,\ \text{where } M = diag(1, -1, 1) \tag{11}$$

where $\hat{t}$ and $\hat{R}$ denote transformed quantities in the right-handed coordinate system. With the above information, we can construct the following transformation matrices:

$$T_{tar \leftarrow glob} = \begin{bmatrix} \hat{R}_{tar}^{glob} & \hat{t}_{tar}^{glob} \\ 0_{1\times 3} & 1 \end{bmatrix}^{\top},\ T_{glob \leftarrow ego} = \begin{bmatrix} \hat{R}_{ego}^{glob} & \hat{t}_{ego}^{glob} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{12}$$

We then obtain the transformation matrix from the ego vehicle coordinate system to the target vehicle coordinate system as:

$$T_{tar \leftarrow ego} = T_{tar \leftarrow glob} \cdot T_{glob \leftarrow ego} \tag{13}$$

To obtain the camera pose matrix in the ego vehicle coordinate system, we construct it as follows:

$$\Phi_{cam} = \begin{bmatrix} \hat{R}_{cam}^{ego} \cdot R_D & \hat{t}_{cam}^{ego} \\ 0_{1\times 3} & 1 \end{bmatrix},\quad R_D = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{14}$$

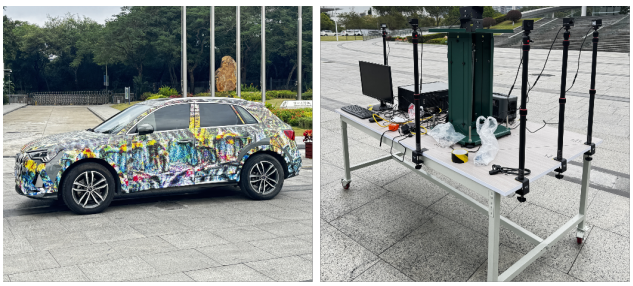
where R_D is a transformation matrix that converts Carla's camera convention (x-forward, z-up) to PyTorch3D's camera convention (z-forward, y-up). Eventually, we compute the camera pose Φ_r in the target vehicle coordinate system, which is required as input by PyTorch3D, via:

$$\Phi_r = T_{tar \leftarrow ego} \cdot \Phi_{cam} \tag{15}$$

For multi-view rendering, We apply the above camera parameter transformation to all six ego-mounted cameras and perform image rendering as defined in Eq 1.

B Physical World Experiments

We have conducted the physical experiment to validate our method in the real world. Figure 5 shows the setup for our physical experiments. We print our generated BEVCA camouflage in a 2D UV map and stick it to a real-life-size Audi vehicle. We also built a simple multi-view camera data collection platform to collect real-life multi-view data. We set up six camera sensors on this platform, following the camera settings (positions and angles) from nuScenes. With this platform, we collect three datasets: one of the car with normal painting, one with CNCA texture, and one of the car with our BEVCA camouflage. Each dataset has 150 samples with different azimuths and distances towards the ego position.



(a) Full size camouflaged car (b) Multi-view camera platform

Figure 5: The physical experiment setup: (a) full size camouflaged car; (b) multi-view camera data collection platform.

Table 4: Performance comparison of BEV-based perception models under normal painting (without attack), BEVCA attack, and after adversarial training. The metrics for 3D detection and BEV segmentation are AP and IoU of the vehicle class, respectively.

METHODS	3D DETECTION					BEV SEGMENTATION		
	BASE	SMALL	TINY	BASE-SEG-DET	SMALL-SEG-DET	LSS	CVT	SINBEVT
NORMAL	0.618	0.721	0.514	0.750	0.634	0.509	0.609	0.570
BEVCA	0.191	0.285	0.381	0.329	0.331	0.253	0.307	0.273
ADVTRAIN	0.615	0.479	0.416	0.683	0.599	0.614	0.567	0.549

C Multi-view Attack Visualization

As shown in Figure 6, we provide some examples of our camouflage attack against 3D object detection in the digital world. Compared with the ground truth label, we can see that the camouflaged vehicle leads to inaccurate bounding boxes or complete "disappearance" from the detector. Our camouflage can achieve successful attacks when the target vehicle appears in not only one but also across views. As shown in Figure 7, we provide examples of our camouflage attack against 3D object detection with full camera views in the real world evaluation.

D Enhancing Model Robustness via Adversarial Training

Beyond identifying model vulnerabilities, our attack also provides a practical mechanism for enhancing the robustness of BEV-based perception models through adversarial training. To demonstrate this, we construct an adversarial dataset consisting of 500 samples generated using our camouflage. To avoid overlap with the evaluation data, the dataset is created under camera configurations different from those used in the test set. We then fine-tune the original BEV perception models on this adversarial dataset. The resulting performance improvements are summarized in Table 4.

The results clearly demonstrate that our adversarial camouflage severely degrades the performance of original BEV-based perception models, exposing critical robustness weaknesses. For instance, BEVFormer-base's detection score drops from 0.618 to 0.191, while CVT falls from 0.609 to 0.307 under attack. Importantly, adversarial training with our adversarial examples substantially restores performance across all models, specifically raising BEVFormer-base to 0.615 and CVT to 0.567, confirming that our method serves not only as a strong attack but also as an effective tool for enhancing model robustness.

E Multi-view robust attack.

To evaluate the robustness of our adversarial camouflage under varying target vehicle positions, we conduct comprehensive experiments across different distances and azimuths relative to the ego vehicle. Specifically, we generate a dataset where the target vehicle appears at distances of [10, 15, 25, 40] meters, aligned with the BEVFormer model's perception range of up to 51.2 meters. For

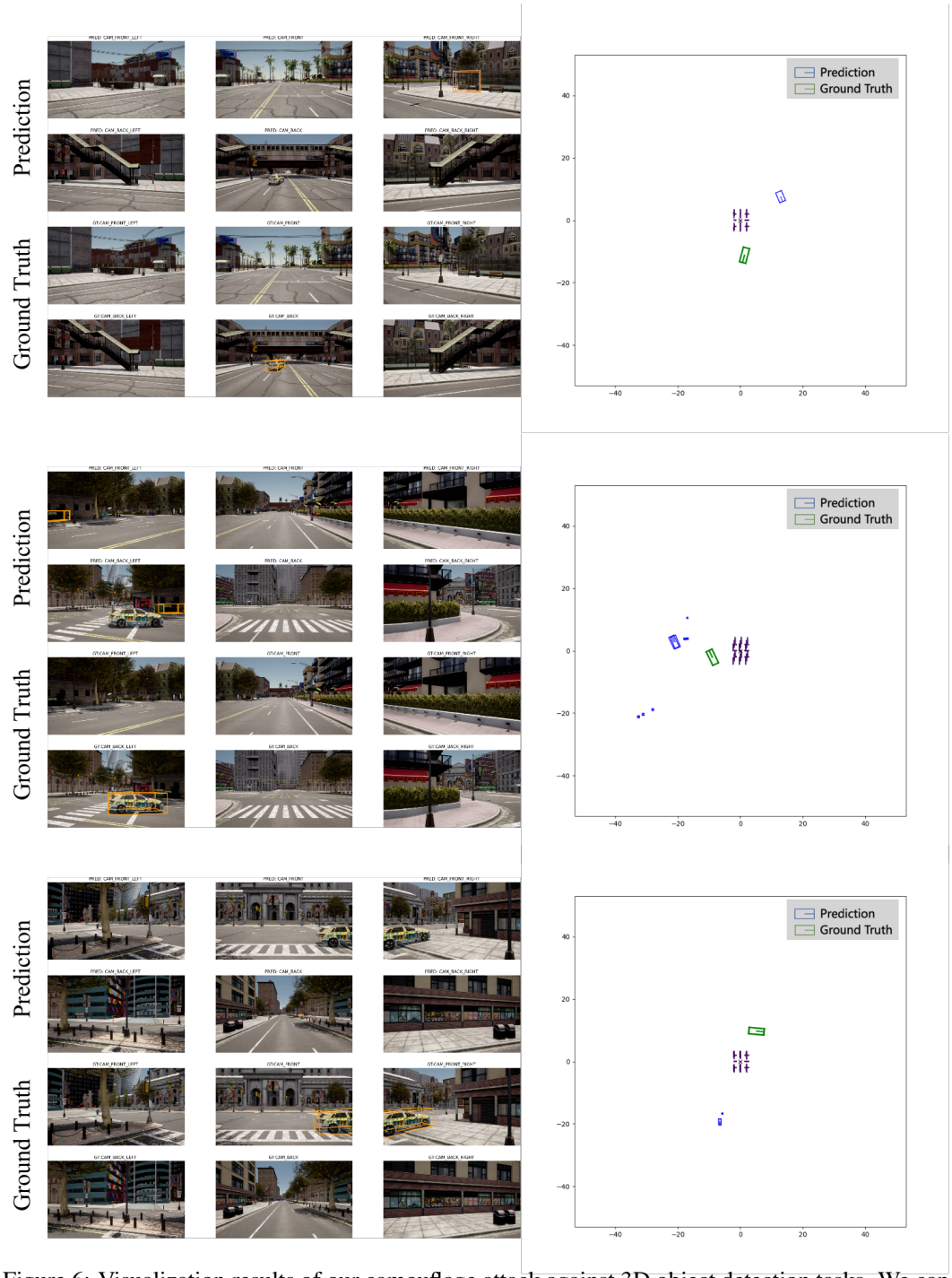


Figure 6: Visualization results of our camouflage attack against 3D object detection tasks. We can observe that the camouflaged vehicle leads to inaccurate bounding boxes or complete "disappearance" from the detector compared to the ground truths.

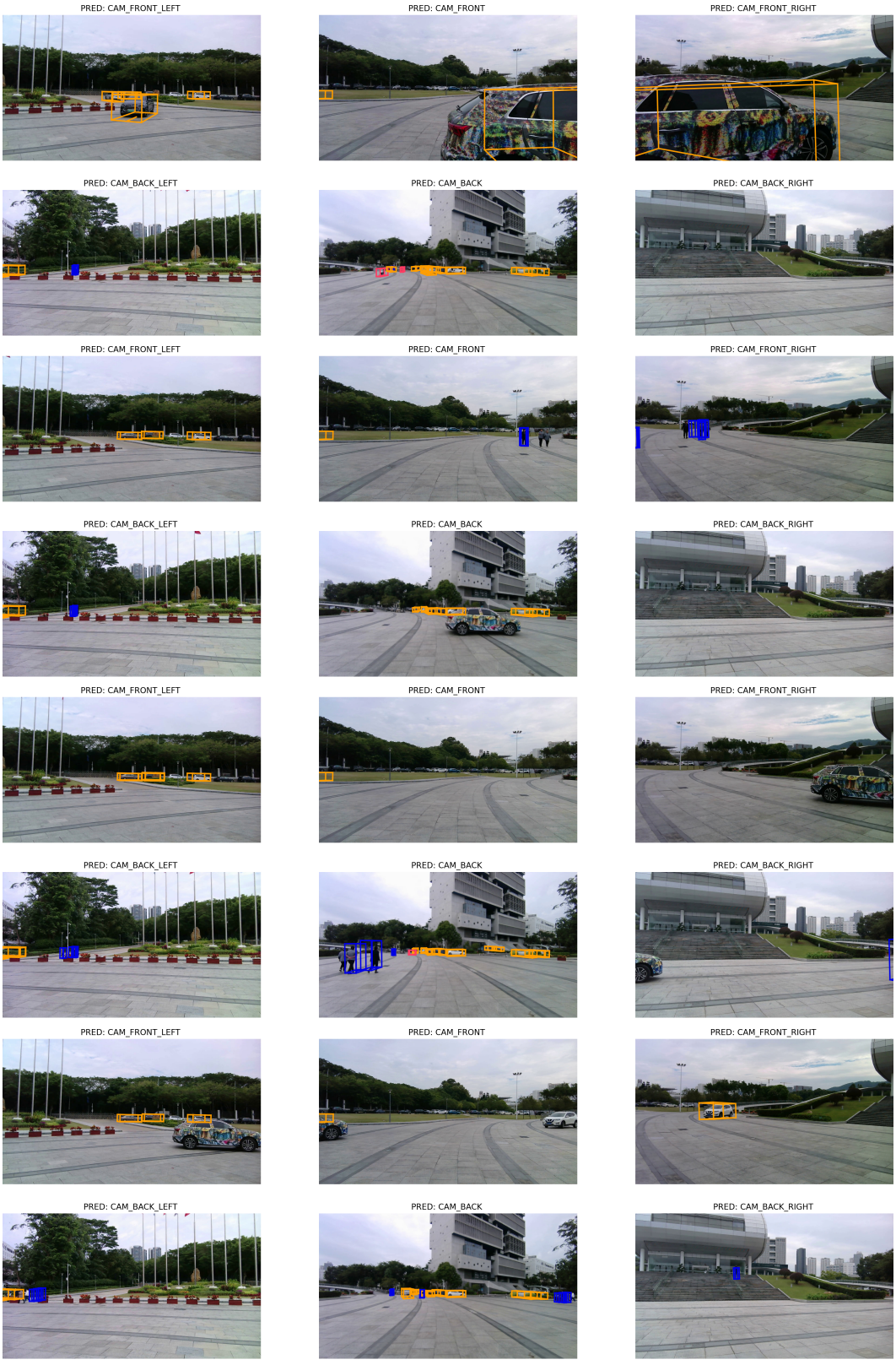


Figure 7: Additional visualization results of our camouflage attack against 3D object detection tasks in the physical world.

Table 5: The attack performance for multi-azimuth and multi-distance. Notice the AP of the normal painted car is 0.618.

AZIMUTH (°)	DISTANCE				AVG
	10	15	25	40	
0	0.278	0.129	0.468	0.376	0.313
60	0.005	0.184	0.311	0.278	0.195
120	0.206	0.332	0.167	0.218	0.231
180	0.123	0.641	0.404	0.001	0.292
240	0.242	0.293	0.300	0.195	0.258
300	0.088	0.282	0.347	0.197	0.228
AVG	0.157	0.310	0.333	0.211	0.253

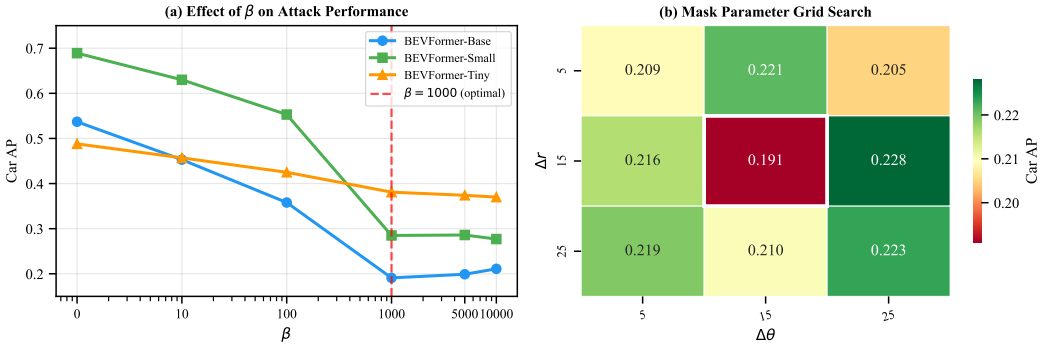


Figure 8: Hyperparameter analysis: (a) Effect of β on attack performance; (b) Grid search over mask parameters Δr and $\Delta\theta$.

azimuths, we sample [0°, 60°, 120°, 180°, 240°, 300°]. For each distance-azimuth combination, we create 40 samples with varying background scenes across different CARLA towns to ensure diversity. We use the BEVFormer base model for evaluation, as other detectors exhibit similar trends.

The results are shown in Table 5. The total mean Car AP value is comparable to the value reported in Table 1. We also observe that with the increase of the distance, the AP first prone to increase at a distance of 10, after that the AP prone to decrease after a distance of 25. This trend can be attributed to two factors: (1) with the increase of the distance, fine-grained adversarial camouflage patterns are difficult for the detection model to identify, causing the decrease of the attack performance; and (2) the detection performance of the model declines at long ranges, causing the AP to decrease again. Nevertheless, our camouflage maintains strong attack effectiveness across all tested configurations, demonstrating its robustness in multi-view settings.

F Effectiveness of hyperparameters during optimization.

We investigate the impact of the main hyperparameters during camouflage optimization related to total variation loss and BEV mask. Specifically, we vary the hyperparameter β that controls the strength of the total variation constraint, and report the results in figure 8 (a) with BEVFormer series models. The results show that incorporating the total variation loss consistently improves attack performance. When β is too small, the generated textures lack coherent structure and fail to deliver strong adversarial signals—especially at long ranges. As β increases, the attack performance improves steadily, highlighting the importance of enforcing total variation in the camouflage pattern. For BEV mask settings, we run hyperparameter tuning experiments with the BEVFormer-base model on two tunable thresholds Δr and $\Delta\theta$ to find suitable values such that it can locate the relevant area accurately. The results from Figure 8 (b) show that Δr with value of 15 and $\Delta\theta$ with values of 15 give the best results. We fix these settings throughout our experiments.

G Robustness Against Simple Defenses.

To contextualize the practical risk of our attack framework, we evaluate BEVCA against three standard, lightweight defensive mechanisms applied to the BEVFormer-base model: BEV Feature Smoothing (Gaussian), Input Camera Image Smoothing (Gaussian), and JPEG Compression.

As shown in Table 6, while aggressive defenses (e.g., severe JPEG compression or heavy smoothing) can marginally degrade the attack's effectiveness, they concurrently introduce non-trivial performance trade-offs on clean data (Normal mAP). BEVCA consistently maintains a massive performance gap over the baseline across all tested defensive parameters, demonstrating robust evasion capabilities that cannot be mitigated by simple input or feature transformations.

Table 6: Performance of BEVFormer-base under BEVCA attack when employing various simple defense mechanisms. The Performance Gap indicates the difference between Normal mAP and BEVCA mAP.

Defense Method	**Parameter**	**Normal mAP**	**BEVCA mAP**	**Performance Gap**
No Defense	-	0.618	0.191	0.427
BEV Feature Smoothing	$\sigma = 0.5$	0.616	0.189	0.427
	$\sigma = 1.0$	0.594	0.175	0.419
	$\sigma = 2.0$	0.482	0.121	0.361
Input Smoothing	$\sigma = 0.5$	0.617	0.171	0.446
	$\sigma = 1.0$	0.614	0.165	0.449
	$\sigma = 2.0$	0.612	0.283	0.329
JPEG Compression	$QF = 75$	0.615	0.150	0.465
	$QF = 50$	0.620	0.136	0.484
	$QF = 25$	0.620	0.113	0.507

H Attack Performance on Original Targeted Models

Table 7 shows the comparison of the effectiveness of different attacks across different 3D detection and BEV segmentation original models, which are trained on the nuScenes dataset [1]. Compared to Table 1, we can notice the performance gap for each attack-model pair between these two tables due to the domain gap between simulation and reality. However, BEVCA still outperforms all the baselines, proving the effectiveness and transferability of our method.

I Usage of LLMs

We have mainly used LLMs to help us with the editing of this paper and write code to anlayze and visulize the experiment resutls. However, the core ideas and methodology of this paper are all original and not influenced by LLMs.

Table 7: Comparison of the effectiveness of attacks across various original 3D detection and BEV segmentation models. The metrics for 3D detection and BEV segmentation are AP and IoU of the vehicle class, respectively.

METHODS	3D DETECTION					BEV SEGMENTATION		
	BASE	SMALL	TINY	BASE-SEG-DET	SMALL-SEG-DET	LSS	CVT	SINBEVT
NORMAL	0.309	0.324	0.243	0.321	0.333	0.209	0.234	0.238
FCA	0.306	0.336	0.263	0.32	0.337	0.204	0.200	0.208
DTA	0.260	0.296	0.249	0.294	0.318	0.175	0.168	0.195
ACTIVE	0.236	0.264	0.220	0.265	0.292	0.178	0.183	0.195
RAUCA	0.259	0.29	0.204	0.295	0.295	0.162	0.157	0.164
CNCA	0.227	0.253	0.185	0.245	0.284	0.157	0.201	0.181
ADVPATCH	0.307	0.310	0.235	0.321	0.331	0.193	0.213	0.202
BEVCA	**0.080**	**0.126**	**0.158**	**0.148**	**0.171**	**0.123**	**0.065**	**0.106**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction explicitly state our core contributions and scope of our paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in a dedicated section 6 .

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: Our work result are mostly experimental and does not involve obtaining theoretical result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have open-sourced our code and described the detailed information to reproduce our experiment in the experiment setting section 4.1.

   Guidelines:

   - The answer [N/A] 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: We have provided the link for the related code and data in the abstract for reproducing our results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We have described our experiment setting and details in the section 4.1.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: We report the mAP, IoU and ASR for the attack performance in the table 1 and 2, which are calculate by dataset s with size at least over than 150. Therefore the reported results have statistical significance.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: We have described the compute resources needed for our experiment in the section 4.1.

Guidelines:

- The answer [N/A] 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: We have reviewd the NeurIPS code of Ethics and confirm our research are conducted aligned with the NeurIPS code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have discussed both the potential positive and negative societal impact in the section 7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We have discussed safeguards to enhance the AI systems against our proposed attack in the section 7.

    Guidelines:

    - The answer [N/A] 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: We confirmed that all the assets that we used in our worked are properly credited. The related license and terms of use explicitly are mentioned and properly respected.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: The new assets introduced in this paper is multi-view neural rendering module and BEV-feature-based adversarial loss. We have provided the detailed documentation for the module and loss function in the section A and 3.3.

Guidelines:

- The answer [N/A] 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: [No]

Justification: We do not involve crowdsourcing and research with human subjects in this paper.

Guidelines:

- The answer [N/A] 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: [No]

    Justification: We do not involve research with human subjects in this paper.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We discuss the usage of LLMs in the section I.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.