SUPPLEMENTARY MATERIAL FOR ADV3D: GENERATING 3D ADVERSARIAL EXAMPLES FOR 3D OBJECT DETECTION IN DRIVING SCENARIOS WITH NERF

This supplementary material is organized as follows:

- We first provide training details of our adversarial examples in Sec. A. Then we evaluate and analyze the transferability of adversarial examples across different detectors in Sec. B, and provide insights to develop more robust 3D detectors in the future.
- In Sec. C, we demonstrate the attack effectiveness of reproducing the adversarial texture in our real world. We then detail the sampling pose distribution $\mathcal{B}$ in Sec. D.
- We provide analysis and ablation studies on the number of adversarial examples in Sec. E, and evaluate the transferability of adversarial texture to the unseen vehicle in Sec. F.
- Finally, we provide repeatability tests in Sec. G, investigate the influence of perceptibility of the attack in Sec. H, and present additional qualitative results in Sec. J. We discuss limitations and future work in Sec. I.

A TRAINING DETAILS

We implement our methods using PyTorch (Paszke et al., 2019) and MMDetection3D (Contributors, 2020). All detectors are resumed from checkpoints available on their open-source repositories to match the original performance exactly. We only select one instance from Lift3D (Li et al., 2023) as the initialization of examples. We conduct our experiments using 8 NVIDIA A100 80G GPUs. We use the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1e-3 for texture latents. In practice, we optimize texture latents on the training set for five epochs with the same batch size as used during training detectors. We do not use any regularization except for semantic-guided regularization. In all experiments without specified, we render two adversarial examples per image.

B TRANSFERABILITY ACROSS DIFFERENT DETECTORS

In Tab. 1, we evaluate the transferability of adversarial examples across different detectors. To this end, we train a single adversarial example of each detector separately, then use the example to evaluate the performance drop of other detectors. We show that there is a high degree of transferability between different models. Among them, we observe that DETR3D (Wang et al., 2021c) appears to be more resilient to adversarial attacks than other detectors. We hypothesize this can be attributed to the sparsity of the query-based method. During the projection of 3D query to the 2D image plane, only a single point of the feature is indexed by interpolation, thus fewer areas of adversarial features will be sampled. This finding may have insightful implications for the development of more robust 3D detectors in the future.

Target \ **Source**	Clean	FCOS3D	PGD-Det	DETR3D	BEVDet	BEVFormer
FCOS3D (Wang et al., 2021a)	0.298	**0.124**	0.141	0.144	0.176	0.158
PGD-Det (Wang et al., 2021b)	0.317	0.172	**0.131**	0.150	0.186	0.172
DETR3D (Wang et al., 2021c)	0.347	0.188	0.170	**0.133**	0.212	0.198
BEVDet (Huang et al., 2021)	0.307	0.148	0.145	0.140	**0.132**	0.140
BEVFormer (Li et al., 2022b)	0.252	0.175	0.155	0.136	0.177	**0.124**

Table 1: Transferability of our attack to unseen detectors. We evaluate the robustness of **target** detectors using an adversarial example trained on **source** detectors. Reported in mAP.

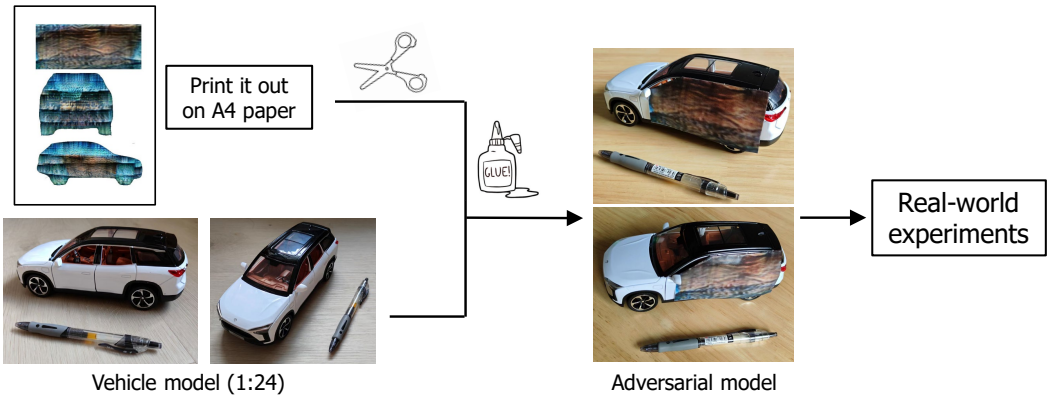

Figure 1: **Setup of real-world experiments.**

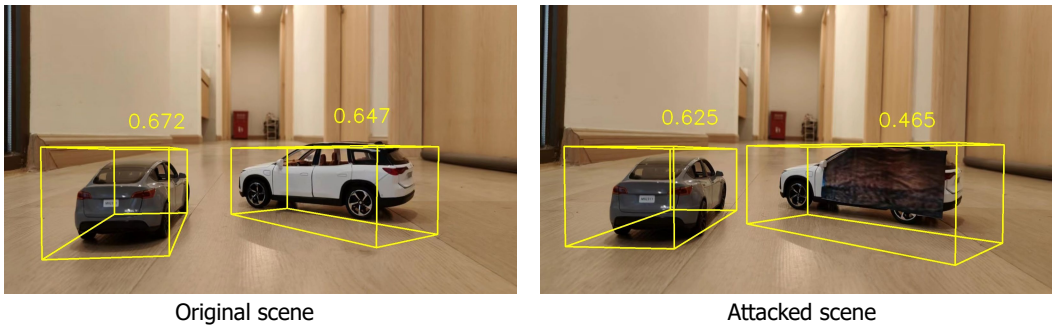

Figure 2: **Results of real-world experiments.** Note that the result is achieved without fine-tuning in the current scene.

C Reproduce Adversarial Examples in Real-World

As depicted in Fig. 1, we reproduce our adversarial scene in the real world by creating a scaled driving scene using 1:24 vehicle models. We approximate the adversarial textures using the rendering of the orthogonal views of examples. We first crop the desired area from the orthogonal view of adversarial example. Next, we print the texture on A4 paper and cut out the desired shape using scissors. We then trim the texture slightly to fit the vehicle model and adhere to it using glue.

After reproducing the adversarial example in our real world, we use it to attack BEVDet (Huang et al., 2021) in our created scene, as shown in Fig. 2. In our experiment, only the front view images are replaced with the attacked images, while the other five surround view images are sampled from the original nuScenes dataset. We plot the predicted 3D boxes and confidences. Due to the inherent domain gap of BEVDet, the pose estimation of vehicles is not satisfactory. Nevertheless, we can observe a drop in confidence of the non-contact object ($0.672 \rightarrow 0.625$) caused by the adversarial texture, proving the practicality of our method. The result can be further improved by fine-tuning a certain adversarial texture in the current scene.

Furthermore, we compare the effect of our method to strong noise. In Tab. 2, we show the experimental results comparing adversarial NeRF with strong noise in real-world settings. In these experiments, we replace the adversarial texture area with various types of noise, including pure black, the mean color of the background image, and random noise. We observe that adversarial NeRF achieves the lowest predicted confidence and outperforms the other three types of texture in terms of attack performance.

Texture	Clean	Pure Black	Mean Color	Random Noise	Adversarial NeRF
Confidence	0.672	0.632	0.634	0.644	**0.625**

Table 2: Comparison of strong noise to our attack.

Experiments	NDS	mAP
Clean	0.3822	0.3076
Trial 1	0.2261	0.1322
Trial 2	0.2263	0.1326
Trial 3	0.2209	0.1310
Trial 4	0.2237	0.1302
Trial 5	0.2228	0.1304
Mean	0.2239	0.1313
Std	0.0023	0.0011

Table 3: Results of repeatability test.

Pose	Distribution	Parameters
x	Uniform	$[-5m, 5m]$
y	Gaussian	$\mu = height, \sigma = 0.2$
z	Uniform	$[10m, 15m]$
l	Gaussian	$\mu = l_{mean}, \sigma = 0.5$
w	Gaussian	$\mu = w_{mean}, \sigma = 0.5$
h	Gaussian	$\mu = h_{mean}, \sigma = 0.5$
θ	Gaussian	$\mu = \pm\pi/2, \sigma = \pi/2$

Table 4: Detailed distribution of pose $\mathbf{b}$ in pose sampling.

D Sampling Pose Distribution

In this section, we introduce the distribution of pose $\mathcal{B}$ used during pose sampling. Specifically, we parameterize the pose $\mathcal{B}$ as 3D bounding boxes that are described by $(x, y, z, l, w, h, \theta)$, where x, y, z is the position of the 3D bounding box's center, l, w, h represent length, width, height of the box, θ is the rotation along y axis. As shown in Tab. 4, we model the components of $\mathcal{B}$ as independently sampled parameters. We mainly follow the parametrization in Lift3D (Li et al., 2023).

To avoid inappropriate sampling in driving scenes (such as a vehicle flying in the sky), we model our objects to roughly be placed on a ground plane in front of the camera. To approximate the ground plane, we set (x, z) to be uniformly sampled from $[-5m, 5m]$ and $[10m, 15m]$. As for y, we first obtain camera height from the camera extrinsic, then we model y as a Gaussian distribution that is centered at the camera height. Similarly, l_{mean}, w_{mean}, and h_{mean} is the mean value of length, width, and height of the 3D box obtained from the statistic of datasets, where $l_{mean} = 4.61m, w_{mean} = 1.95m, h_{mean} = 1.72m$. Then we model the size of 3D boxes as a Gaussian distribution that is centered at the mean value. The rotation of the object is considered as a bimodal distribution that is centered at forward facing($\pi/2$) and backward facing($-\pi/2$).

E Ablation of The Number of Adversarial Examples

Since the number of populated adversarial examples is related to the area of adversarial texture, we ablate the number of rendered adversarial examples per image in Tab. 7. We observe that the more adversarial objects, the better the attack performance. The number 0 denotes a clean evaluation without attacks. In all experiments in our main paper except Sec. 5.2, we render two adversarial examples for evaluation to avoid overcrowded scenes. We note that for number > 1, the rendered ad-

Experiments	NDS	mAP
Clean	0.3822	0.3076
SUV	0.2209	0.1310
SUV to Sedan	0.2344	0.1449
SUV to Hatchback	0.2319	0.1413
SUV to Jeep	0.2348	0.1452

Table 6: Results of our adversarial texture transfer to unseen shape.

Number	NDS	mAP
0 (clean)	0.3822	0.3076
1	0.2648	0.1895
2	0.2247	0.1325
3	0.1749	0.0698

Table 7: Ablation of the number of adversarial examples.

versarial patches are from the same adversarial example, since we only maintain a single adversarial object during training.

F Adversarial Texture Transfer to Unseen Vehicle

To further validate the transferability of our method, we analyze the attack performance of a trained adversarial texture transfer to an unseen shape. As shown in Tab. 6, we train our adversarial examples using only an SUV-type vehicle and then transfer the adversarial texture to other unseen vehicles by our proposed disentangled texture and shape generation. The target types of vehicles include Sedan, Hatchback, and Jeep. The transfer results of the unseen types of vehicles show comparable attack performance compared with the seen one, which demonstrates the transferability of our adversarial texture to unseen shapes. The rendered images of adversarial examples can be found in Fig. 4-6.

G Repeatability Tests

In our pose sampling, we randomly and independently sample the parameters of the pose for each frame from pose distribution $\mathcal{B}$. In this section, we investigate the impact of the randomness of pose sampling. We conduct five independent evaluations using the same trained adversarial example and show results in Tab. 3. We found that regardless of the random pose sampling, all the trials display a similar attack performance. The results indicate that our adversarial attack is robust against the randomness of pose sampling.

H Impact of the Perceptibility of the Attack

We conduct experiments investigating the influence of distance and pixel proportion. In the setting of 3D vision, the farther the distance, the smaller the pixel proportion (size of patch / (1600*900)), thus less perceptible. From Tab. 8, we can observe that as the distance increases, the pixel proportion decreases and the attack performance relatively decreases as well. This indicates that the higher the perceptibility (larger pixel proportion), the better the attack performance.

Distance (m)	**10**	**11**	**12**	**13**	**14**
Pixel proportion (%)	4.67	3.67	2.96	2.42	2.04
Performance drop (%)	41.47	39.56	37.11	34.74	32.52

Table 8: Influence of perceptibility to our attacks

I Limitation and Future Work

Learning to Sample and Attack As we do not have access to the dataset annotations, we can not model the explicit relationship between adversarial and normal objects to avoid collision, and the collision itself can cause a performance drop ("No Parts" in Tab.3 of the main paper). Future work can apply geometry-aware composition (Chen et al., 2021) to mitigate this problem. Additionally, future research can explore learning to predict optimal poses of adversarial objects to maximize the effectiveness of attacks.

Broader Impact The recent development of NeRF has led to remarkable progress in NeRF-based driving scene simulation (Yang et al., 2023; Li et al., 2023; 2022a). Our adversarial framework is general and can be extended to integrate with the advances in NeRF-based simulators to benefit a wide spectrum of practical systems. For instance, our framework can be combined with UniSim (Yang et al., 2023) to perform adversarial closed-loop evaluations of self-driving cars in NeRF environments, or with ClimateNeRF (Li et al., 2022a) to identify adverse weather conditions that may corrupt the autonomous driving system. We believe that our work provides valuable insights and opens up new possibilities for creating authentic adversarial evaluations that improve the robustness of self-driving cars.

Potential Harmful Consequences The trained adversarial examples have the potential to induce serious traffic accidents in driving scenarios. However, our work is not intended to cause disruptions in autonomous driving systems. Instead, our goal is to use the examples to gain a deeper understanding of the systems and improve their robustness. We hope our work will draw more attention of the community to further verify and enhance the robustness of autonomous driving systems.

J Additional Qualitative Results

In this section, we present additional qualitative results of our trained adversarial example. In Fig. 3, we show rendered images of an SUV-type adversarial example, and different parts of semantic-guided regularization. Then we display rendered images of adversarial texture transfer to the unseen vehicles in Fig. 4-6.

Figure 3: **First row:** the initial SUV-type vehicle. **Second row:** the trained full-part adversarial example using the SUV-type vehicle. **Third row to the fifth row:** The results of full part adversarial texture transfer to the front part, side part, and rear part.

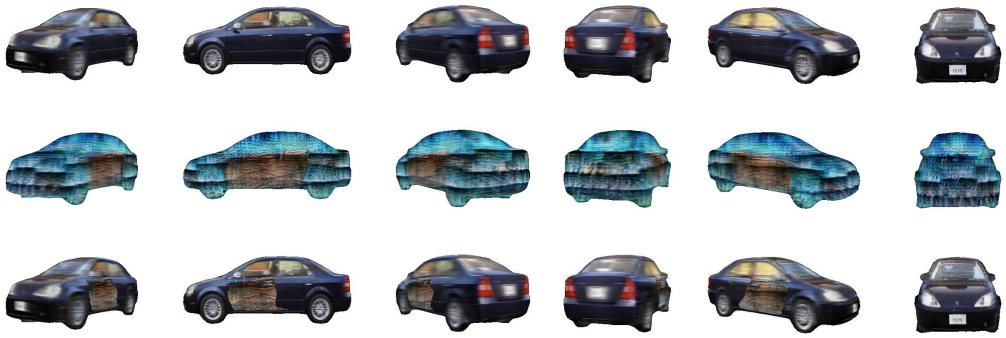

Figure 4: **First row:** Rendered images of a Sedan-type vehicle. **Second and third rows:** We transfer the trained adversarial texture of an SUV-type vehicle (Fig. 3) to the Sedan-type vehicle.

Figure 5: **First row:** Rendered images of a Hatchback-type vehicle. **Second and third rows:** We transfer the trained adversarial texture of an SUV-type vehicle (Fig. 3) to the Hatchback-type vehicle.

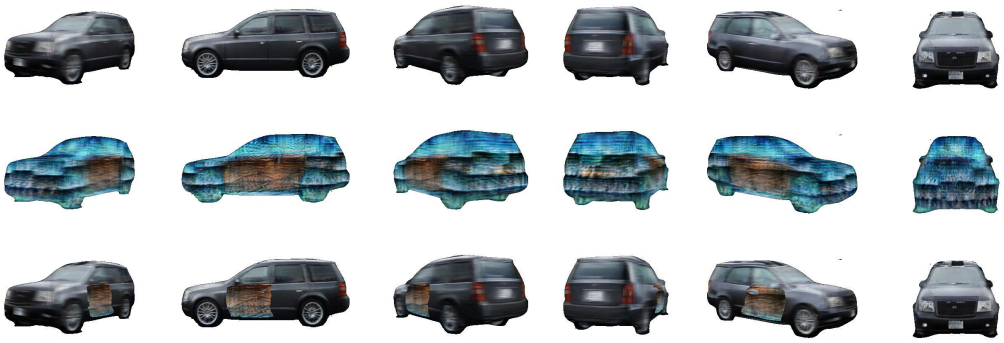

Figure 6: **First row:** Rendered images of a Jeep-type vehicle. **Second and third rows:** We transfer the trained adversarial texture of an SUV-type vehicle (Fig. 3) to the Jeep-type vehicle.

REFERENCES

Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *CVPR*, 2021.

MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Leheng Li, Qing Lian, Luozhou Wang, Ningning Ma, and Ying-Cong Chen. Lift3d: Synthesize 3d training data by lifting 2d gan to 3d generative radiance field. In *CVPR*, 2023.

Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Physically-based neural rendering for extreme climate synthesis. *arXiv e-prints*, pp. arXiv–2211, 2022a.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *ECCV*, 2022b.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCV Workshop*, 2021a.

Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting objects in perspective. In *CoRL*, 2021b.

Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2021c.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.