SABER: Spatially Consistent 3D Universal Adversarial Objects for BEV Detectors — Supplementary Material —

Aixuan Li[1], Mochu Xiang[1], Bosen Hou[1], Zhexiong Wan[1], Jing Zhang[2], Yuchao Dai[1] *
[1]School of Electronics and Information, Northwestern Polytechnical University &
Shaanxi Key Laboratory of Information Acquisition and Processing, Xi'an, China
[2] School of Computing, Australian National University

In this Supplementary Material, we first describe the implementation details in Sec. 1 and illustrate the Realistic Occlusion checking process in Sec. 2. Secondly, we provide extended ablation studies in Sec. 3, covering both loss function components and the contributions of color and geometry, to further justify our design choices. We then provide additional results on black-box transferability in Sec. 4 and further detail the setup and digital validation of our physical attacks in Sec. 5. Moreover, we present extended analyses of the distance robustness of our attack mesh in Sec. 6, as well as results on attacking camera-only models trained with LiDAR supervision in Sec. 7. To examine whether our attack can be mitigated by defenses, we also evaluate its effectiveness against a robust model in Sec. 8. Finally, we provide additional metric comparisons and visualization results of our proposed non-invasive attack mesh in Sec. 9.

1. Implementation details

Optimization and Training Setup: We keep mesh topology fixed and optimize both vertex positions and texture using Adam (learning rate 0.02), with per-vertex displacement capped at 0.1 meter. To avoid overlapping, the offset is adjusted as the mesh size changes. The mesh location is determined from the current frame and kept fixed across temporal inputs. Training uses eight NVIDIA L40 48G GPUs for 10 epochs. The training time is 14.1h (BEVDet), 15.1h (BEVDet4D), and 57.4h (BEVFormer). Due to memory constraints, the number of meshes per image is set to 4 for BEVDet/BEVDet4D and 2 for BEVFormer, whereas 15 meshes are used at test time. We set α and β to 10 in Eq.(9) of main text.

Details for Shape Initialization Experiment: In the experiment regarding the initialization of 3D objects with different shapes, all meshes are placed with a 0.1m offset from the target's bottom-right corner to avoid invasive. The *cylinder* mesh contains 1642 vertices and 3240 faces, with a base radius of 0.3 and a height of 2.0. The *cube* mesh contains 726 vertices and 1200 faces, with each edge measuring 0.9 meter. The *sphere* mesh consists of 2562 vertices and 5120 faces, with a radius of 0.5.

2. Realistic Occlusion check

In Sec. 3.3 of the main paper, the determination of realistic occlusion follows a two-step procedure, which involves checking the overlap of 2D bounding boxes and the overlap of convex hulls in the BEV space. To clearly illustrate this procedure, Fig. 1 visualizes the determination scheme for Occ_{2D} (Eq. (2) in the main paper) and the rationale behind Occ_{BEV} (Eq. (3) in the main paper). Furthermore, Fig. 1 (c) highlights the limitations of relying solely on depth for occlusion checking. Specifically, when the vehicle depth is smaller than that of the adversarial mesh, a naive depth-based approach would mistakenly infer that the vehicle occludes the mesh. The proposed convex-hull-based method effectively resolves this ambiguity.

3. Extended Ablation Studies

3.1. Ablation for loss function

To validate if $\mathcal{L}_{\text{sim}}$ effectively encourages false positives in irrelevant regions, we conduct ablation studies on different loss components in Tab. 1. Specifically, we employ mAP for evaluation, as it is the primary metric most sensitive to false detections in nuScenes. The further decrease in mAP upon adding $\mathcal{L}_{\text{sim}}$ confirms its effectiveness in inducing scene-level confusion and security risks, justifying its necessity.

Table 1. **Ablation study of loss functions on BEVDet.**

$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{loc}}$	$\mathcal{L}_{\text{sim}}$	Clean mAP	Init mAP	Adv mAP
✓			0.3086	0.2625	0.1506
✓	✓		0.3086	0.2625	0.1311
✓	✓	✓	0.3086	0.2625	0.1298

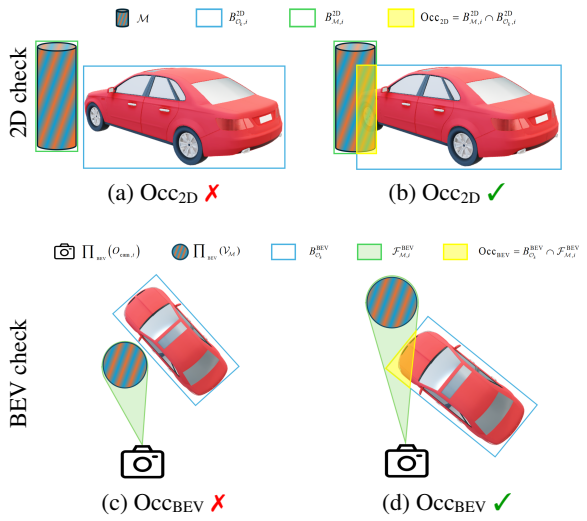


Figure 1. **Illustration of the Realistic Occlusion checking process.** The top row visualizes the occlusion check in 2D space, while the bottom row depicts the check in BEV space. The yellow regions indicate the overlapping areas. The evaluation of Occ_{BEV} is triggered only when $\text{Occ}_{\text{2D}} > 0$. The adversarial mesh is considered occluded by a scene object only when both the Occ_{2D} and Occ_{BEV} conditions hold.

3.2. Ablation for Color and Geometry

We provide additional ablation studies to compare the independent effects of color and geometry optimizations with cylinder initialization in Tab. 2. Results show that color-based optimization is more effective, aligning with [2] in main paper that solely optimizing geometry yields limited adversarial effects.

Table 2. **Ablation of color and geometry optimization.**

Geometry	Color	Clean NDS	Init NDS	Adv NDS	Clean mAP	Init mAP	Adv mAP
✓	-	0.3942	0.3579	0.3505	0.3086	0.2625	0.2519
-	✓	0.3942	0.3579	0.2136	0.3086	0.2625	0.1357
✓	✓	0.3942	0.3579	0.2097	0.3086	0.2625	0.1298

4. Results of 3D Adversarial Objects under Transfer Attacks

To validate the cross-model generalization capability of our 3D adversarial mesh, we conduct transfer-attack evaluations on both full scenes and the "vehicle" category using BEVDet, BEVDet4D, and BEVFormer (Tab. 3 and Tab. 4). The results indicate that the 3D adversarial objects consistently exploit common weaknesses across these BEV models, independent of the scenario or target category.

5. Physical Attack Setup and Digital Validation

To evaluate the proposed adversarial attack method under real-world physical conditions, we construct a compact physical BEV data acquisition setup. In principle, a BEV perception system employs six synchronized cameras mounted around a vehicle to provide 360° surround perception [1]. Due to hardware limitations, we emulate this configuration using a single ZED2i stereo camera. Specifically, we capture six views sequentially by placing the camera at fixed positions corresponding to the vertices of a regular hexagon marked on the ground, each separated by 60°. In addition, six auxiliary images are captured from slightly shifted viewpoints to assist in accurate extrinsic reconstruction, although these auxiliary images are not used for detection experiments. Our data acquisition setup and the camera pose diagram are illustrated in Fig. 2 Camera intrinsics are obtained through standard chessboard calibration [10]. The initial extrinsics are estimated using VGGT [7] from the multi-view images and subsequently refined based on the physically measured camera positions and manual inspection to recover the real-world scale. This physical data-capture procedure effectively emulates a fixed six-camera surround-view configuration and produces geometrically consistent inputs for BEV-based 3D object detection.

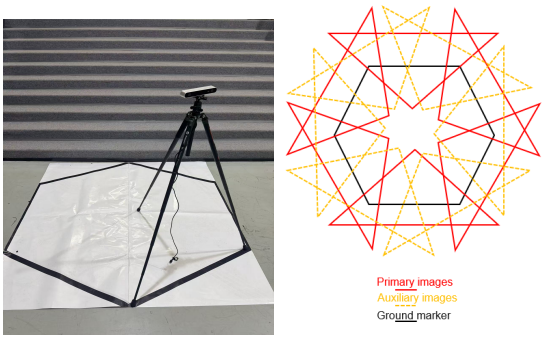
To transfer a learned digital adversarial mesh into the real world by printing, we apply two pragmatic simplifications during mesh optimization. First, we only optimize the mesh **texture**, not the mesh geometry, which eases fabrication. Secondly, we adopt a **two-stage training approach**. The initial stage consists of 10 epochs under a standard digital setting. The second stage, lasting 5 epochs, introduces physically-aware rendering: we modify the rendering light based on scene weather (night, clear, rain) and add random perturbations. The base light intensities for night, clear, and rain are set to [0.45, 0.8, 0.5], with corresponding perturbation ranges of [0.1, 0.2, 0.15], respectively. During this second stage, we also apply a masked Total Variation

Table 3. **Performance comparison of attack on the full scene by 3D adversarial objects under transfer attacks.** The first row shows the source mesh model, the first column shows the victim model.

victim \ source	Init		BEVDet	BEVDet4D	BEVFormer
	NDS	mAP	NDS / mAP	NDS / mAP	NDS / mAP
BEVDet	0.3579	0.2625	0.2097 / 0.1298	0.2306 / 0.1422	0.2869 / 0.1856
BEVDet4D	0.4158	0.2734	0.3229 / 0.1716	0.2762 / 0.1564	0.3309 / 0.1868
BEVFormer	0.4592	0.3402	0.4049 / 0.2829	0.4039 / 0.2821	0.2876 / 0.1652

Table 4. **Performance comparison of attack on the "Vehicle" category by 3D adversarial objects under transfer attacks.** The first row shows the source mesh model, and the first column shows the victim model.

victim \ source	Init			BEVDet			BEVDet4D			BEVFormer		
	mAP_{vel}	$\text{AP}_{car0.5}$	$\text{AP}_{car2.0}$	mAP_{vel}	$\text{AP}_{car0.5}$	$\text{AP}_{car2.0}$	mAP_{vel}	$\text{AP}_{car0.5}$	$\text{AP}_{car2.0}$	mAP_{vel}	$\text{AP}_{car0.5}$	$\text{AP}_{car2.0}$
BEVDet	0.187	0.1489	0.5835	0.038	0.0141	0.2048	0.050	0.0342	0.2872	0.101	0.0629	0.4366
BEVDet4D	0.195	0.1513	0.5936	0.068	0.0422	0.3214	0.051	0.0226	0.2461	0.095	0.0739	0.4581
BEVFormer	0.280	0.1607	0.6890	0.206	0.1190	0.5627	0.204	0.1190	0.5578	0.101	0.0474	0.3590



(a) (b)

Figure 2. **Overview of our data acquisition setup.** (a) Hardware setup: a camera mounted on a tripod used for image capture.(b) Diagram illustrating the 12 camera poses: 6 primary images for model input and 6 auxiliary images. The auxiliary images are captured from slightly shifted viewpoints to assist in accurate extrinsic reconstruction.

(TV) loss [6] to the mesh's texture. This loss, denoted as $\mathcal{L}_{\text{TV}}(I, M)$, is guided by the rendered image I (corresponding to I_s^{rgb}) and the mask M (corresponding to I_s^{mask}) from Eq.(5) in main text, and is defined as:

$$\mathcal{L}_{\text{TV}}(I, M) = \lambda \left(\frac{\sum_{i,j} \left[(I_{i+1,j} - I_{i,j})^2 \cdot M'_h \right]}{\sum_{i,j} M'_h + \epsilon} + \frac{\sum_{i,j} \left[(I_{i,j+1} - I_{i,j})^2 \cdot M'_w \right]}{\sum_{i,j} M'_w + \epsilon} \right), \tag{1}$$

where $M'_h = M_{i+1,j} \cdot M_{i,j}$ and $M'_w = M_{i,j+1} \cdot M_{i,j}$.

Using the calibrated multi-view captures and the print-aware mesh optimization, we perform physical experiments on driving scenes with the BEVDet model [1]. As summarized in Tab. 5, we first confirm in a controlled, digital setting that the simplified geometry and print-aware color modeling still produce strong adversarial effects.

Extending this to the real world, Fig. 3 presents visual examples of our physical deployment, where the printed meshes cause consistent degradation in BEV detection across multiple scenes, viewpoints, and visibility conditions. We observe the following primary effects of our physical attack: (1) Localization Errors: The attack causes severe bounding box shifts (mis-localization) and, in some cases, can even lead to complete detection suppression (*i.e.* the target vehicle's prediction box vanishes). (2) Cross-view Corruption: The attack successfully transfers to views where the attack mesh is not directly visible, corrupting the predictions in these unobserved views. (3) False Positives: The attack generates multiple spurious bounding boxes in the target region. (4) Misclassification: The model predicts an incorrect category for the target object. (5) Occlusion Robustness: The attack remains effective even when the adversarial mesh is partially occluded by environmental obstacles (*e.g.* vehicle).

Another interesting finding is that, under similar viewing angles, different orientations of our mesh can produce varying attack effects, as shown in Fig. 3 (c) and (d). This also highlights the advantage of using an adversarial object, as its inherent 3D nature allows it to be consistently observed and optimized across various views and partial occlusions during training.

Through physical attack experiments, we verify that our printed adversarial mesh successfully mislead BEV detection in real-world scenes. This demonstrates that our approach not only works in digital simulation but also transfers effectively to physical deployment, highlighting the potential real-world security risks of BEV perception systems.

Physical-to-digital gap. While our physical attack remains effective, we observe a performance gap compared to digital simulations. This discrepancy primarily stems from color fidelity degradation caused by the gamut misalignment between digital RGB textures and ink-printed CMYK outputs. To mitigate this, potential future directions include: (1) establishing a differentiable color mapping model for local printers to compensate for printing distortions during the optimization process, (2) using cylindrical electronic displays instead of static ink printing.

Table 5. **Digital-domain evaluation of our physical-ready mesh against the purely digital baseline, without Real Occ.**

	Clean NDS	Init NDS	Adv NDS	Clean mAP	Init mAP	Adv mAP
BEVDet_dig	0.3942	0.3579	0.2097	0.3086	0.2625	0.1298
BEVDet_phy	0.3942	0.3579	0.2139	0.3086	0.2625	0.1368

6. Distance Robustness Analysis

6.1. Generalization to untrained target-to-object distances

In **Attack distance** of the main text, we compare the results of training 3D adversarial objects at different locations. To further assess spatial robustness, we place a mesh trained at 0.5 units across varying distances (Tab. 6). The adversarial object successfully attacks the target category across all test locations, indicating robustness over distance. While the attack is more effective near the training position, ASR gradually declines as the placement moves farther away. This also explains why the ASR in the Sec.4.4 **Placement Generalization** of the main text is lower when the number of meshes is small: insufficient adversarial object coverage causes the adversarial mesh to be distant from the target vehicle, thereby compromising the attack performance.

6.2. Attacks on vehicles at varying target-to-ego distances

To investigate the attack performance relative to the distance between the target and ego vehicles, we stratified the evaluation results into discrete distance intervals. As detailed in

(a) Distant mis-localization (b) Cross-view corruption (c) False positive generation (d) Misclassification (e) Occlusion Robustness

Figure 3. **Visualization of our adversarial attack's effectiveness and properties.** (a) Distant mis-localization: The attack induces significant localization errors (bounding box shifts) for vehicles at a distance. (b) Cross-view corruption: A gray/adversarial mesh is placed in the right-side camera view. This attack successfully transfers to the left-side view, where the mesh is not visible, corrupting its prediction and demonstrating the attack's cross-view capability. (c) False Positive Generation: The attack induces the model to erroneously detect an additional bounding box (white car) on the right side, predicting two vehicles instead of one. (d) Misclassification: The model erroneously detects the barrier (physical adversarial mesh itself) as two pedestrians. (c) and (d) demonstrate different failure modes in the same scene and location but with varying mesh orientations. (e) Occlusion Robustness: the attack remains effective even when the adversarial mesh is partially occluded by vehicle.

Table 6. **Comparison of attack performance across different adversarial object distances from the target vehicle**, using a 3D adversarial object trained at a fixed distance of 0.5 m with Real Occ.

Distance	Init NDS	Adv NDS	Init mAP	Adv mAP	$ASR_{0.3}$	$ASR_{0.5}$
Clean	0.3942	-	0.3086	-	-	-
0.1	0.3682	0.2682	0.2754	0.1610	0.452	0.510
0.3	0.3684	0.2708	0.2772	0.1612	0.446	0.511
0.5	0.3719	0.2710	0.2796	0.1609	0.444	0.511
0.7	0.3707	0.2729	0.2789	0.1622	0.439	0.504
1.0	0.3700	0.2782	0.2786	0.1667	0.448	0.494
1.5	0.3739	0.2807	0.2812	0.1689	0.445	0.488
2.0	0.3722	0.2796	0.2791	0.1711	0.436	0.489
2.5	0.3736	0.2888	0.2819	0.1778	0.422	0.473
3.0	0.3749	0.2944	0.2855	0.1839	0.404	0.462

Tab. 7, our proposed universal mesh attack maintains a high Attack Success Rate against the BEVDet model across all evaluated ranges, from near to far. This result demonstrates that our method exhibits robust and consistent efficacy across diverse distance scales.

Table 7. **Comparison of attack performance at varying distances between the target vehicle and the ego vehicle,** using $ASR_{0.3}$ as the evaluation metric.

	0-20 m	20-40 m	40-60 m
Ours	0.366	0.781	0.929
Adv3D	0.300	0.490	0.536

7. Attacking Camera-Only Models Trained with LiDAR Supervision

While our analysis in the main text primarily focuses on attacks against purely camera-based BEV models, some existing methods [3, 4, 8, 9] utilize LiDAR information as supervision during the training phase while operating as camera-only models at inference. We selected [9] as a representative method from this category to validate the effectiveness of our attack, as shown in Tab. 8. We observe a more pronounced decline in mAP compared to NDS. This suggests that our attack method is more adept at inducing missed detections rather than degrading the attribute predictions, such as position, size, and orientation, for True Positives. This is further corroborated by the high Attack Success Rate (ASR) maintained across various IoU thresholds.

Table 8. **Performance comparison of our adversarial attack on GeoBEV [9]**, a representative camera-only model trained with LiDAR supervision.

	Real Occ	Clean NDS	Init NDS	Adv NDS	Clean mAP	Init mAP	Adv mAP	$ASR_{0.3}$	$ASR_{0.5}$	$ASR_{0.7}$
GeoBEV [1]	✗	0.5459	0.5058	0.3493	0.4296	0.3732	0.1640	0.704	0.767	0.808
GeoBEV [1]	✓	0.5459	0.5174	0.4068	0.4296	0.3906	0.2259	0.447	0.512	0.579

8. Attack Against Robust Model

With the efficacy of our method on standard models previously established, we proceed to investigate whether our attack remains effective against models fortified with defense strategies. To this end, we performed experiments on BEVDet by incorporating adversarial training to train a robust model. Specifically, we implemented data augmentation by replacing 10% of the total training data with adversarial examples generated via Projected Gradient Descent (PGD) [5]. We then fine-tuned the standard BEVDet model on this augmented dataset for two epochs to obtain the robust BEVDet model. Subsequently, following the same pipeline as described in the main text, we inserted the adversarial object optimized on the standard BEVDet model into the scenes

to evaluate the performance of this robust model. For the PGD configuration, we set the step size to $1/255$, the maximum perturbation budget to $8/255$, and the number of steps to 20. As shown in Tab. 9, despite the defense strategies, the robust model experiences a substantial decline in both NDS and mAP, accompanied by a high ASR. This demonstrates that our proposed method is capable of bypassing simple defense measures.

Table 9. **Evaluating attack effectiveness against a defense-enhanced (robust) BEVDet model.** "Clean" refers to the baseline performance on the test set, while "Ours" reports the results of attacking this robust model using our adversarial mesh optimized on the standard (non-robust) BEVDet model.

	Real Occ	Init NDS	Adv NDS	Init mAP	Adv mAP	$ASR_{0.3}$	$ASR_{0.5}$
Clean		0.3779	-	0.2835	-	-	-
Ours	✗	0.3418	0.2339	0.2372	0.1472	0.511	0.547
Ours	✓	0.3518	0.2791	0.2514	0.1764	0.346	0.380

9. Additional Results for White-Box Attacks

Considering the varying object sizes in the nuScenes dataset, we follow the official nuScenes evaluation protocol, which does not directly utilize IoU. Tab. 10 reports the mean Average Precision (mAP) for the vehicle category and detection results of cars at different sensor ranges. For example, $AP_{car0.5}$ denotes the average precision for vehicles within 0.5 meters of the sensor. The results demonstrate that our 3D adversarial samples consistently reduce vehicle detection accuracy, particularly within 0.5 meters, highlighting both the vulnerability of current BEV models and the potential for severe impact in close-range scenarios. In addition, we show more visualized comparisons in Fig. 4 and Fig. 5.

Table 10. **Performance comparisons using additional evaluation metrics** for BEV-based 3D object detection under adversarial attacks with Real Occ.

	Init/Adv mAP_{vel}		Init/Adv $AP_{car0.5}$		Init/Adv $AP_{car1.0}$		Init/Adv $AP_{car2.0}$		Init/Adv $AP_{car4.0}$	
BEVDet	0.205	0.073	0.1684	0.0470	0.4163	0.1924	0.6300	0.3413	0.7481	0.4432
BEVDet4D	0.216	0.085	0.1761	0.0390	0.4282	0.1779	0.6465	0.3303	0.7535	0.4662
BEVFormer	0.297	0.154	0.1790	0.0822	0.4521	0.2976	0.7142	0.5398	0.8367	0.6835

References

[1] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 3, 4

[2] Yao Huang, Yinpeng Dong, Shouwei Ruan, Xiao Yang, Hang Su, and Xingxing Wei. Towards transferable targeted 3d adversarial attack in the physical world. In *IEEE CVPR*, pages 24512–24522, 2024. 2

[3] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, pages 1477–1485, 2023. 4

[4] Zhenxin Li, Shiyi Lan, Jose M Alvarez, and Zuxuan Wu. Bevnext: Reviving dense bev frameworks for 3d object detection. In *IEEE CVPR*, pages 20113–20123, 2024. 4

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 4

[6] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE CVPR*, pages 5188–5196, 2015. 3

[7] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE CVPR*, pages 5294–5306, 2025. 2

[8] Junyin Wang, Chenghu Du, Huikai Liu, Zhenchang Xia, Bingyi Liu, and Shengwu Xiong. Hybridbev: Hybrid encode and distillation for improved bev 3d object detection. *IEEE T-ITS*, 26(11):21257–21270, 2025. 4

[9] Jinqing Zhang, Yanan Zhang, Yunlong Qi, Zehua Fu, Qingjie Liu, and Yunhong Wang. Geobev: Learning geometric bev representation for multi-view 3d object detection. In *AAAI*, pages 9960–9968, 2025. 4

[10] Z. Zhang. A flexible new technique for camera calibration. *IEEE TPAMI*, 22(11):1330–1334, 2000. 2

Figure 4. **Visualizations of attack effects in image view.**

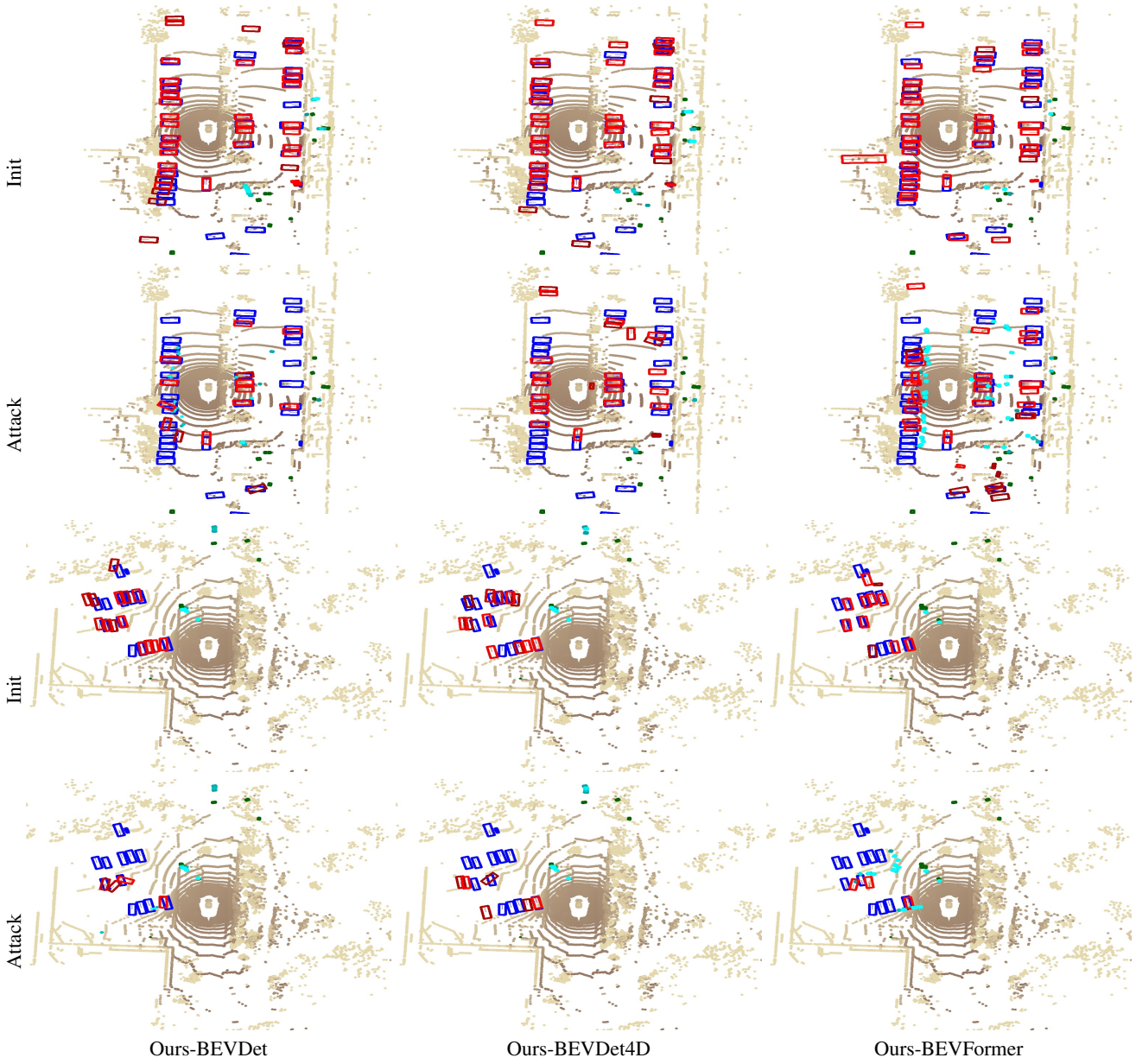


Figure 5. **Visualizations of attack effects in the BEV.** Top: predictions with initial objects. Bottom: predictions after inserting the adversarial object. Blue/red indicate ground-truth/predicted boxes for vehicles; green/cyan for other object categories.