Supplementary Material for SHIFT3D: Synthesizing Hard Inputs For Tricking 3D Detectors

Hongge Chen[1], Zhao Chen[1], Gregory P. Meyer[1], Dennis Park[1],
Carl Vondrick[1], Ashish Shrivastava[1], and Yuning Chai[1]

[1]Cruise LLC

A. Detailed Information of Rendering and Gradient Calculation

A.1. Rendering

This section details how we cast rays towards SDF objects that have been inserted into a point cloud scene. As noted in the main paper, the first step is to determine the object's bounding box center and draw a sphere with a 7-meter radius centered at that point. When generating adversarial poses, we restrict displacement to less than 4 meters on the xy-plane to ensure that no portion of the object extends beyond this 7-meter sphere.

A LiDAR point $\hat{\boldsymbol{x}}_i$ in the input scene is evaluated for potential adjustment (due to now being occluded by the inserted object) only if a straight line (the laser beam) connecting it to the LiDAR source intersects with the 7-meter sphere centered at the object insertion location. We refer to these points as $\{\hat{\boldsymbol{x}}_i | i \in \mathcal{I}_{\text{calc}}\}$. Any LiDAR points that do not meet these criteria are assumed to remain unoccluded after the object insertion, and are thus kept unchanged ($\boldsymbol{x}_i \leftarrow \hat{\boldsymbol{x}}_i$). Figure 8 illustrates this process.

To process the LiDAR points in $\{\hat{\boldsymbol{x}}_i | i \in \mathcal{I}_{\text{calc}}\}$, we sample $J+1$ points on all beams. To calculate the step size for the i^{th} beam, we first compute its beam length $k_i = \|\hat{\boldsymbol{x}}_i - \boldsymbol{s}\|_2$. Then, the step size for this beam is set to be k_i/J. Using this step size, we sample points $\hat{\boldsymbol{x}}_i^{(j)}$ along the beam between the point, $\hat{\boldsymbol{x}}_i$, and the LiDAR source, $\boldsymbol{s}$, where $j = 0, 1, \ldots, J$, and endpoints $\hat{\boldsymbol{x}}_i^{(0)} = \boldsymbol{s}$ and $\hat{\boldsymbol{x}}_i^{(J)} = \hat{\boldsymbol{x}}_i$. The value J is chosen to make sure that none of the beams have a step size greater than 0.1 meters.

Next, we evaluate the signed distance function $g(\boldsymbol{z}, T(\hat{\boldsymbol{x}}_i^{(j)}; \boldsymbol{\theta}))$ for all the sampled points. If for all j, $g(\boldsymbol{z}, T(\hat{\boldsymbol{x}}_i^{(j)}; \boldsymbol{\theta})) > 0$, then $\hat{\boldsymbol{x}}_i$ is not occluded and we set $\boldsymbol{x}_i \leftarrow \hat{\boldsymbol{x}}_i$ and $m_i \leftarrow 0$.

Otherwise, $\hat{\boldsymbol{x}}_i$ is occluded and needs to be replaced by a $\boldsymbol{x}_i$ on the surface of the inserted SDF object, and $m_i \leftarrow 1$. To this end, we find the first j such that $g(\boldsymbol{z}, T(\hat{\boldsymbol{x}}_i^{(j)}; \boldsymbol{\theta})) < 0$, which is denoted as j_{in}. Note that since we place the object at least 15m away from the AV, we know that $j_{\text{in}} > 0$ always holds. Finally, we run a binary search of 30 steps between $\hat{\boldsymbol{x}}_i^{(j_{\text{in}})}$ and $\hat{\boldsymbol{x}}_i^{(j_{\text{in}}-1)}$ to find the $\boldsymbol{x}_i$ such that $|g(\boldsymbol{z}, T(\boldsymbol{x}_i; \boldsymbol{\theta}))| < \epsilon = 10^{-6}$. $\boldsymbol{x}_i$ is on the surface of the SDF object and added to the output scene, while the original $\hat{\boldsymbol{x}}_i$ is occluded and removed from the output scene.

A.2. Pose Transformation

To describe the pose transformations $T(\cdot; \boldsymbol{\theta})$, we use a 6-dimensional vector $\boldsymbol{\theta}$ consisting of the coordinates $(x_{\boldsymbol{\theta}}, y_{\boldsymbol{\theta}}, z_{\boldsymbol{\theta}})$ and the angles of yaw ($\delta_{\boldsymbol{\theta}}$), pitch ($\beta_{\boldsymbol{\theta}}$), and roll ($\gamma_{\boldsymbol{\theta}}$). The transformation matrix is given as

$$T(\boldsymbol{x}; \boldsymbol{\theta}) = R(\delta_{\boldsymbol{\theta}}) R(\beta_{\boldsymbol{\theta}}) R(\gamma_{\boldsymbol{\theta}}) \left[\boldsymbol{x} - (x_{\boldsymbol{\theta}}, y_{\boldsymbol{\theta}}, z_{\boldsymbol{\theta}})^T\right], \quad (9)$$

where $R(\delta_{\boldsymbol{\theta}},)$, $R(\beta_{\boldsymbol{\theta}})$, and $R(\gamma_{\boldsymbol{\theta}})$ are 3×3 3D rotation matrices of yaw, pitch, and roll.

A.3. Gradient Calculation Using Automatic Differentiation Tools

To compute Eq. (7) using automatic differentiation tools, we first calculate $(\boldsymbol{e}_i \cdot \frac{d\mathcal{L}}{d\boldsymbol{x}_i})$ and $(\boldsymbol{e}_i \cdot \frac{\partial g(\boldsymbol{z}, \cdot)}{\partial \boldsymbol{x}_i})^{-1}$ for all $\boldsymbol{x}_i$, utilizing two back-propagation operations on $\mathcal{L}$ and g. Subsequently, we detach the resulting two tensors from the computational graph and treat them as coefficients. Finally, we perform another back-propagation operation with respect to $\boldsymbol{z}$ on a weighted sum of SDF values. A similar three-step strategy is also applied to Eq. (8).

B. Detailed Information of Experiments

B.1. Metrics of Detectors

We first train our PointPillars and SST detector models exclusively on point cloud data from the Waymo Open Dataset training set. We did not incorporate any auxiliary inputs such as intensity. We show the performance of our model on the vanilla WOD detection task in Table 4 and compare it with the baseline model in [20].

B.2. Hyper-Parameters

In Section 4.3, we provided a summary of the hyper-parameters utilized in our experiments. Here, we provide additional details regarding the hyper-parameters used in the generation of adversarial shape and pose.

We conducted a hyper-parameter search to determine the optimal learning rate α for generating both adversarial shape and adversarial pose. Specifically, we considered

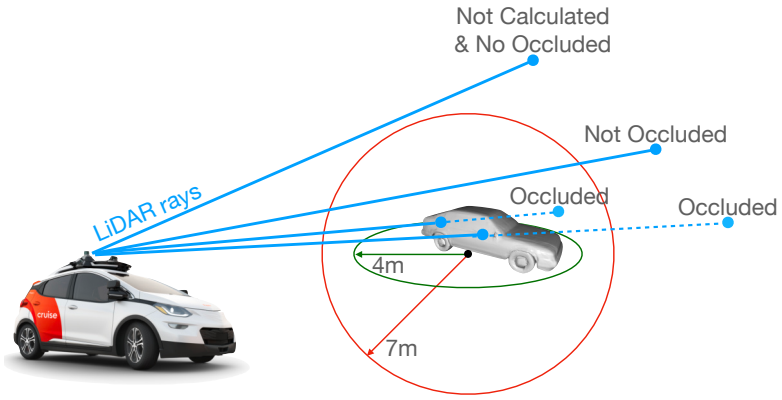


Figure 8: Rendering an SDF object with realistic occlusion by moving background points to the object's surface. We evaluate points only if the beam from the LiDAR source to that point intersects with a 7-meter radius sphere drawn around the inserted object. When changing the object's pose, we limit displacement to under 4 meters on the xy-plane so that there is no part of the object exceeding the 7-meter sphere.

Model	Metric (Vehicle)	Overall BEV (LVL_1/LVL_2)	Overall 3D (LVL_1/LVL_2)
Baseline in [20]	APH AP	79.1/71.0 80.1/71.9	62.8/55.1 63.3/55.6
Our PointPillars	APH AP	81.0/73.3 82.3/74.6	60.3/52.6 61.0/53.3
Our SST	APH AP	89.2 /81.8 90.0/ 82.5	74.8/66.4 75.3/66.8
Our PointPillars after Fine-tuning	APH AP	81.2/73.5 82.6/74.9	60.5/52.8 61.2/53.5

Table 4: Our PointPillars and SST detectors achieves comparable APH and AP for vehicles as the baseline model reported in [20]. The fine-tuned PointPillars detector model shown in the last row is discussed in Section 4.7.

values of α in the set $\{0.0001, 0.001, 0.01, 0.1\}$. In addition, for adversarial shape generation experiments, we also searched for the optimal value of λ in the set $\{0.1, 1, 10\}$.

To select the optimal value of α, we employed the threshold-recall curve's AUC metric as the target. Our experiments revealed that $\alpha = 0.01$ resulted in the lowest AUC value for both shape and pose experiments. Moreover, for adversarial shape generation, the lowest AUC value was obtained with $\alpha = 0.01$ regardless of the value of λ.

To select the optimal value of λ, we chose the value that produced the lowest loss value $\mathcal{L}_{\text{adv}}(\boldsymbol{z})$. In instances where different values of λ produced similar loss values, we chose the larger value of λ to minimize the perturbation. Our experiments determined that $\lambda = 10$ was optimal for Coupe and Sports Car, while $\lambda = 1$ was optimal for SUV, Convertible Car, and Beach Wagon.

C. Additional Visualizations for Adversarial Shape Generation

We randomly select 50 adversarial shapes generated by SHIFT3D and present them in Figure 9. We group them by the baseline objects used for generation. We observe that most of the shapes generated are semantically meaningful.

We also present the adversarial objects and the corresponding detection scores at various steps of the optimization process in Figure 10. Notably, we observe that SHIFT3D object detection scores smoothly decrease, which demonstrates that SHIFT3D can be used to explore challenging examples for a detection model in a continuous space of increasing difficulty, rather than at just a singular point.

D. Additional Qualitative Results for Adversarial Pose Generation

In Figure 11 we present more visualizations for the baseline and challenging poses produced by SHIFT3D in their scenes. Interestingly, we observe that the detection model tends to fail when the inserted SHIFT3D vehicle is closed to or partially occluded by other objects in the scenes, such as trees, bushes, fences, or other vehicles.

E. Experiments on SST detectors

In this section, we present the results of generating adversarial shape and pose on an SST detector, in order to demonstrate that SHIFT3D can be used with various net-

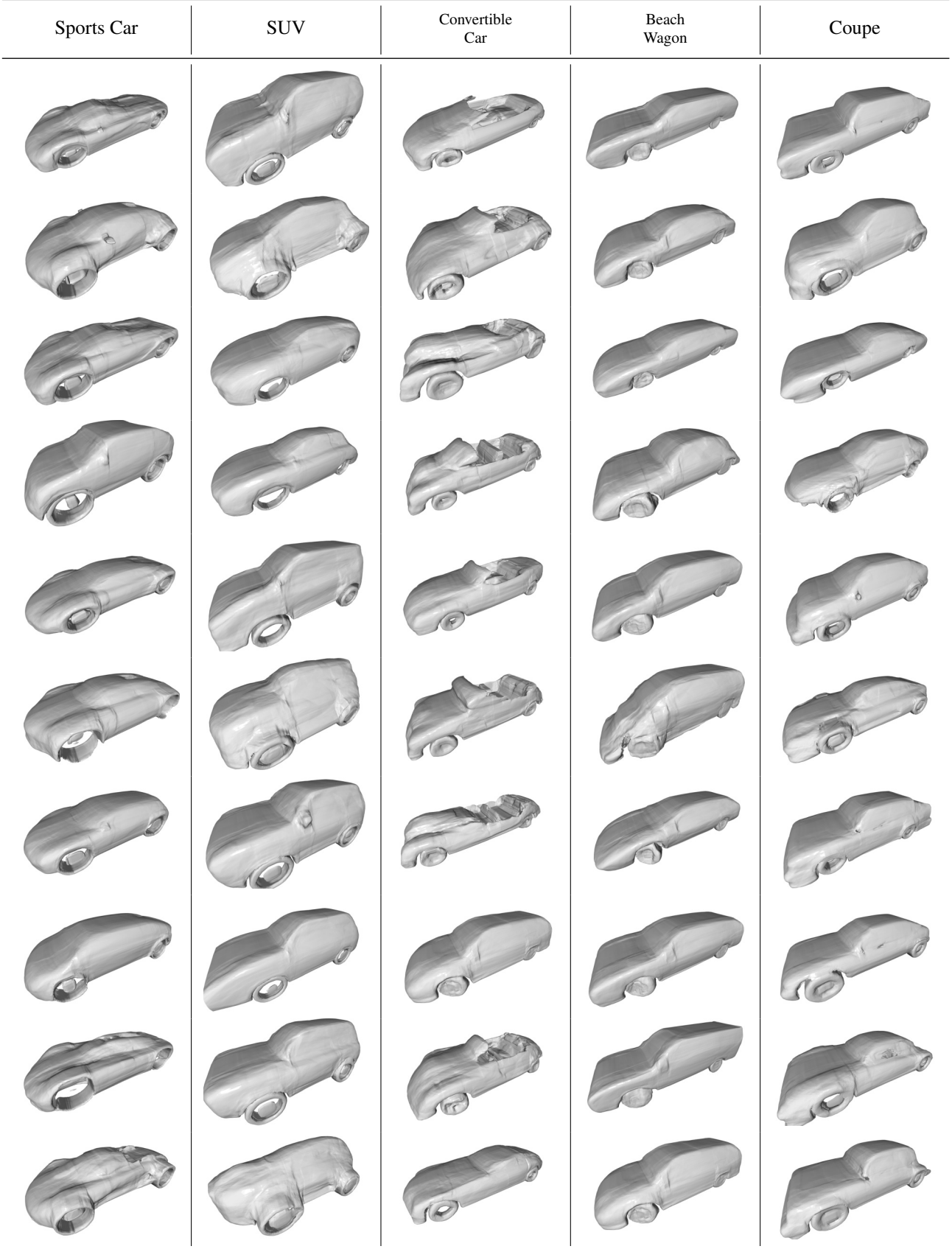
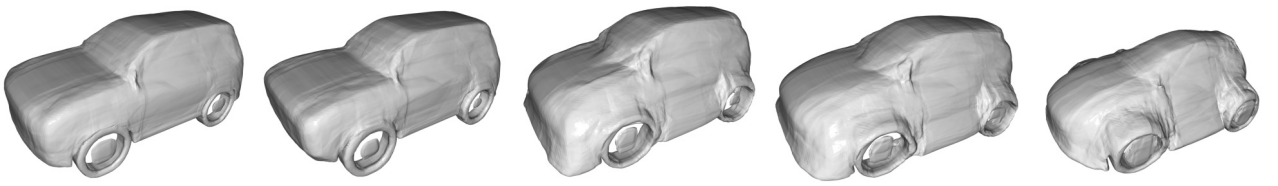


Figure 9: Visualizations of random selected challenging objects generated by SHIFT3D



(1.a) Score = 0.82 (1.b) Score = 0.60 (1.c) Score = 0.43 (1.d) Score = 0.35 (1.e) Score = 0.03

(2.a) Score = 0.58 (2.b) Score = 0.42 (2.c) Score = 0.35 (2.d) Score = 0.17 (2.e) Score = 0.00

Figure 10: Adversarial shapes and their corresponding detection scores at different steps of the optimization process. (a) shows the initial shape corresponding to the baseline object, while (b)-(d) depict intermediate shapes at various stages of the optimization. (e) shows the final adversarial shape at the end of the optimization.

work architectures. Our SST detector's performance metrices on natural WOD vehicles are shown in Table 4. We can see that it overperforms the baseline in [20].

E.1. Adversarial Shape Generation Results

Figure 12 presents the adversarial shape generation results for an SST detector, with threshold-recall curves for each baseline object and its corresponding adversarial objects across the 500 scenes. To quantify the overall reduction in recall, we compute the area under the threshold-recall curves (AUC) and report the corresponding numerical values in Table 5.

Method	Area Under Curve (AUC)				
	Coupe	Sports Car	SUV	Conv. Car	Beach Wagon
Baseline	0.812	0.817	0.743	0.726	0.830
SHIFT3D	**0.533**	**0.524**	**0.471**	**0.448**	**0.585**
SHIFT3D evaluated w/ PointPillars	**0.559**	**0.538**	**0.581**	**0.461**	**0.623**

Table 5: The AUC for the curves in Figure 12, which demonstrates that SHIFT3D produces challenging shapes that confuse an SST detector and SHIFT3D shows high transferability.

E.2. Adversarial Pose Generation Results

Our results, presented in Figure 13 and Table 6, demonstrate a significant reduction in recall performance for adversarial poses, even when placed in alternate poses.

Method	Area Under Curve (AUC)				
	Coupe	Sports Car	SUV	Conv. Car	Beach Wagon
Baseline	0.812	0.817	0.743	0.726	0.830
SHIFT3D	**0.681**	**0.678**	**0.518**	**0.544**	**0.740**
SHIFT3D evaluated w/ PointPillars	**0.691**	**0.695**	**0.659**	**0.601**	**0.739**

Table 6: The AUC for the curves in Figure 13, which demonstrates that SHIFT3D produces challenging poses that confuse an SST detector and SHIFT3D has high transferability.

E.3. SHIFT3D Transferability between PointPillars and SST

In order to investigate the transferability of SHIFT3D between different detector models, we also test our PointPillars detector on the shapes and poses generated by SHIFT3D with the SST detector. As observed in Figures 12 and 13 and Tables 5 and 6, both shape and pose generated by SHIFT3D show high transferability, even between two completely different model structures.

F. Retrieving the Nearest Match for SHIFT3D Queries in Natural Objects

To test the semantic features produced by SHIFT3D on a set of natural objects, we examine objects within the WOD validation set that closely resemble the output of SHIFT3D. These experiments will show that SHIFT3D is not only useful for understanding 3D object detectors, but also for data discovery within large, possibly unlabeled datasets.

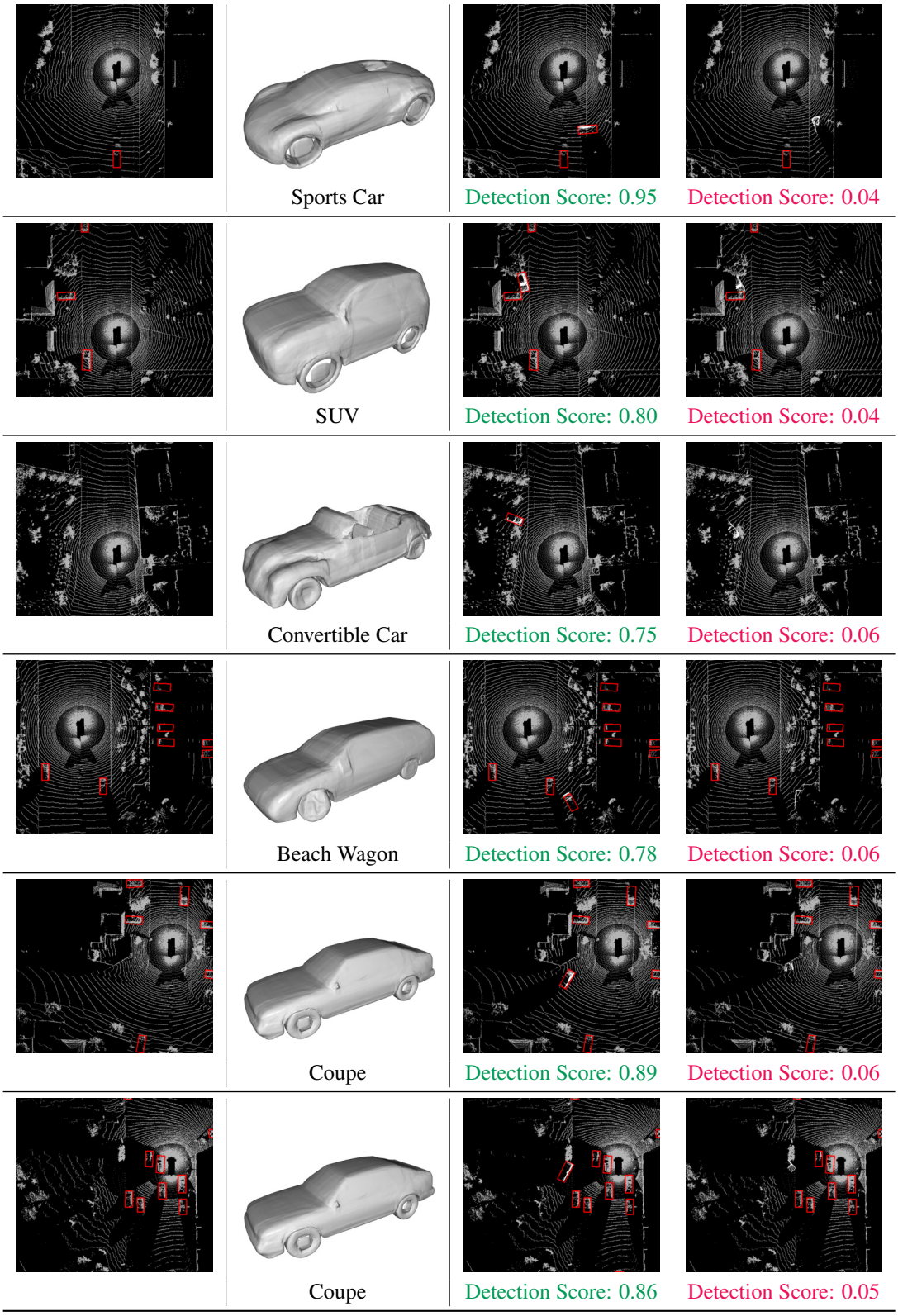
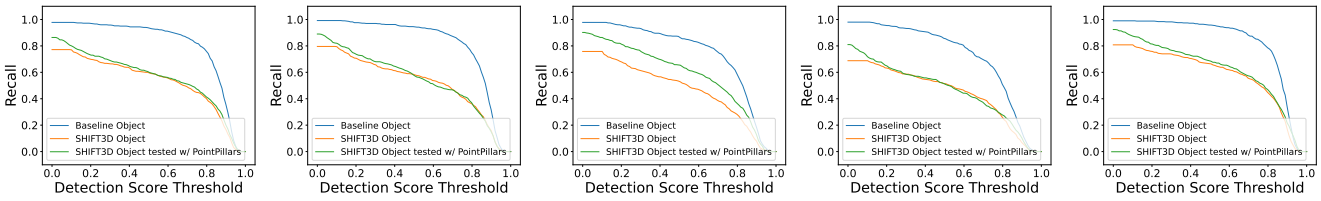
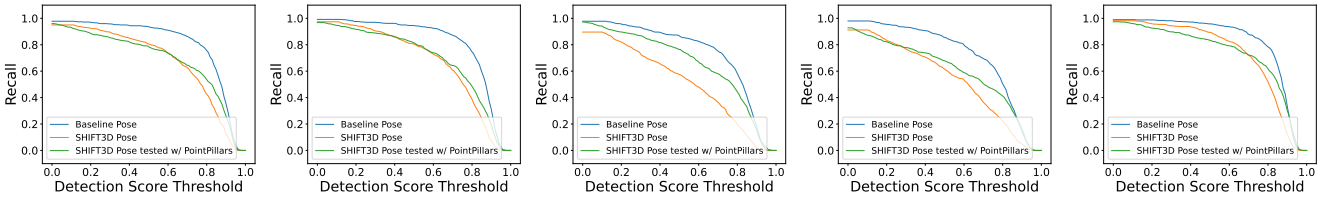


Figure 11: Additional visualizations of challenging poses and their scenes. Red boxes are detector's output.



(a) Coupe (b) Sports Car (c) SUV (d) Convertible Car (e) Beach Wagon

Figure 12: Threshold-recall curves to evaluate adversarial *shape* generation for different vehicles in the "Automobile" category with the SST object detector. Each curve represents the recall rate of the detector at different detection threshold values, computed from 500 scenes with the corresponding vehicle present. SHIFT3D demonstrates its effectiveness in deceiving the SST detector. We also evaluated the PointPillars's performance on these objects generated with the SST detector to show the transferability of SHIFT3D.



(a) Coupe (b) Sports Car (c) SUV (d) Convertible Car (e) Beach Wagon

Figure 13: Threshold-recall curves evaluate adversarial pose for "Automobile" category vehicles with the SST detector. Similar to the curves in Figure 12, each curve shows the SST detector recall rate at different thresholds, from 500 scenes with the vehicle present. SHIFT3D exhibits significantly lower recall rates and high transferability.

First, we focus on reconstructing the shapes of natural vehicles from the WOD validation set from their partially occluded LiDAR point clouds. This process begins by cropping out the point cloud corresponding to each vehicle. Using DeepSDF, we then reconstruct the vehicle's shape. The goal is to optimize the shape latent parameters, $\boldsymbol{z}$, so that all points are nearly equal to 0. However, a direct optimization of $\boldsymbol{z}$ in its native 256-dimensional space often leads to overfitting, resulting in shapes that are not realistic. To counter this, we adopt a dimensionality reduction strategy, akin to the methods in [22] and [7]. We perform a PCA on the vehicle shapes' $\boldsymbol{z}$ in our DeepSDF training set to create a 10-dimensional subspace Π_{PCA}.

Mathematically, for a set of points $\{\boldsymbol{x}_i\}$ within a bounding box, where the ground truth pose is $\boldsymbol{\theta}$, shape reconstruction is achieved by:

$$\boldsymbol{z}_{\text{natural}} = \underset{z \in \Pi_{\text{PCA}}}{\arg\min} \sum_i \|g(\boldsymbol{z}, T(\boldsymbol{x}_i; \boldsymbol{\theta}))\|^2. \tag{10}$$

Subsequently, a retrieval pool is formulated by calculating $\{\boldsymbol{z}_{\text{natural}}\}$ for 8000 natural vehicles in the WOD. For an object generated by SHIFT3D, represented by $\boldsymbol{z}_{\text{SHIFT3D}}$, we identify the natural object whose $\boldsymbol{z}_{\text{natural}}$ is the most similar by minimizing the ℓ_2 distance:

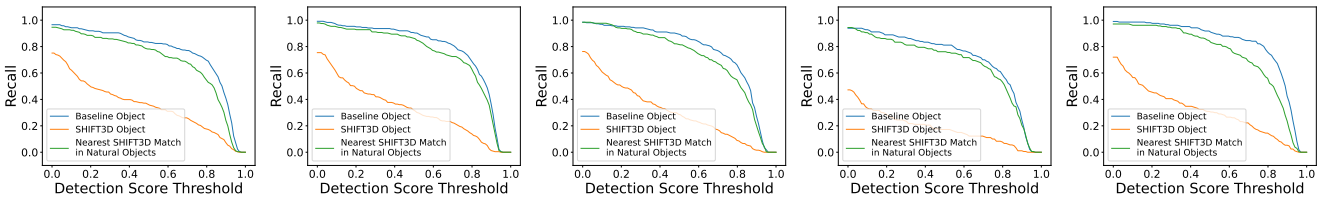
$$\boldsymbol{z}^*_{\text{natural}} = \underset{z_{\text{natural}}}{\arg\min} \|\boldsymbol{z}_{\text{natural}} - \boldsymbol{z}_{\text{SHIFT3D}}\|_2 \tag{11}$$

To guarantee the reliability of the retrieval, we exclude instances where the computed z^*_{natural} is significantly different from $\boldsymbol{z}_{\text{SHIFT3D}}$. For clarity, in our study, only objects with $\|\boldsymbol{z}_{\text{natural}^*} - \boldsymbol{z}_{\text{SHIFT3D}}\|_2 < \|\boldsymbol{z}_{\text{SHIFT3D}}\|_2$ are considered.

For evaluation purposes, we position the retrieved objects in identical poses to the SHIFT3D objects to eliminate the influence of pose variation. The results, depicted as a threshold-recall curves, can be found in Figure 14. Additionally, visual comparisons between SHIFT3D objects and their natural counterparts are showcased in Figure 15.

We note that some of the retrieved objects (e.g. in the latter columns) do look visually similar to our SHIFT3D object, but some (e.g. the former columns) look quite visually different. We attribute this discrepancy to the limited nature of the retrieval dataset; SHIFT3D will often generate examples that are not close to any example in the retrieval dataset, either because the retrieval dataset is small or not diverse enough, or because the SHIFT3D object falls outside the real data distribution. However, generating OOD samples is very much one of the clear advantages of SHIFT3D, as we can use these objects that are rare or impossible to find in the real world to learn lessons about our models' failure modes that would be hidden from us otherwise. More crucially, the retrieved objects always produce lower detection scores from our detector than the baseline objects. In other words, despite some discrepancies in visual similarity, us-

ing SHIFT3D to retrieve real objects seems to consistently provide us with interesting examples that tend to fool our detector. And this application of SHIFT3D is agnostic to these objects being labeled, so it can be applied to large, unlabeled datasets.



(a) Coupe (b) Sports Car (c) SUV (d) Convertible Car (e) Beach Wagon

Figure 14: Threshold-recall curves to evaluate the retrieved nearest matches of adversarial *shape* generation for different vehicles in the "Automobile" category. We only plot objects such that $\|z_{\text{natural}^*} - z_{\text{SHIFT3D}}\|_2 < \|z_{\text{SHIFT3D}}\|_2$

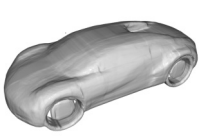
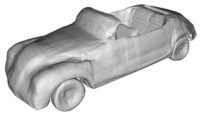
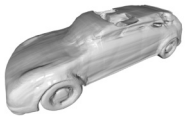
	Sports Car	SUV	Convertible Car	Beach Wagon	Coupe
Baseline Objects	 Score: 0.91	 Score: 0.41	 Score: 0.62	 Score: 0.50	 Score: 0.60
SHIFT3D Objects	 Score: 0.04	 Score: 0.0	 Score: 0.14	 Score: 0.10	 Score: 0.11
Retrieved Natural Objects	 Score: 0.82	 Score: 0.07	 Score: 0.23	 Score: 0.40	 Score: 0.33

Figure 15: Visualizations of the retrieved nearest matches of SHIFT3D objects.