Seeing is Not Believing: Adversarial Natural Object Optimization for Hard-Label 3D Scene Attacks

Supplementary Material

Table 8. Comparisons on attack performance against 3D grounding models on the ScanRefer dataset.

3D Model	Attack	Unique		Multiple	
		Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
3D SPS [62]	[7]	43.27	37.89	34.36	30.73
	[24]	35.75	30.19	28.08	25.52
	[40]	41.06	36.34	32.65	29.44
	Ours	**19.29**	**17.87**	**12.94**	**10.43**
EDA [84]	[7]	45.40	39.19	37.04	33.38
	[24]	38.21	31.64	30.03	27.92
	[40]	44.15	37.72	34.80	31.59
	Ours	**20.61**	**17.54**	**15.38**	**12.82**

Table 9. Does the adversarial object have different adversarial effects when placed in different positions?

Distance to align target center	Overall		Chamfer Distance	Hausdorff Distance
	Acc@0.25	Acc@0.5		
0.0	15.96	13.73	0.0003	0.0021
1.0	16.22	14.17	0.0005	0.0028
2.0	16.41	14.25	0.0008	0.0034
3.0	16.74	14.38	0.0014	0.0045

6. Performance Comparison with Other Methods

As shown in Table 8, we try to re-implement existing 2D hard-label attack methods into the 3D domain for comparison. We can find that our method is more effective.

7. Impact of the Location of the Adversarial Object Injection

In all our experiments, we directly align the position of the object trigger on the selected plane with the original object center to achieve an efficient attack. Moreover, since our attack goal is to cause the mislocalization (to trigger position) of the grounding model, placing the trigger near the target object also has a more disruptive effect on the target. Of course, we can initialize the adversarial object in different positions. As shown in Table 9, we vary the distances between the visual trigger and the target object center, and corresponding results indicate that our attack is still effective in fooling the model in different locations. It also shows that a larger distance just slightly degenerates the attack performance and requires relatively larger perturbations.

Table 10. Attack effect of the adversarial object at various positions relative to the target object.

Position relative to target object	Overall	
	Acc@0.25	Acc@0.5
Our selected position	15.96	13.73
Position on the target object	16.08	13.72
Position around the target object (left)	17.35	14.86
Position around the target object (right)	17.27	15.11
Position around the target object (front)	17.14	14.90
Position around the target object (behind)	17.09	15.03
Random position far away from target object	18.41	15.94

8. Effect of the Adversarial Object at Various Positions Relative to the Target Object

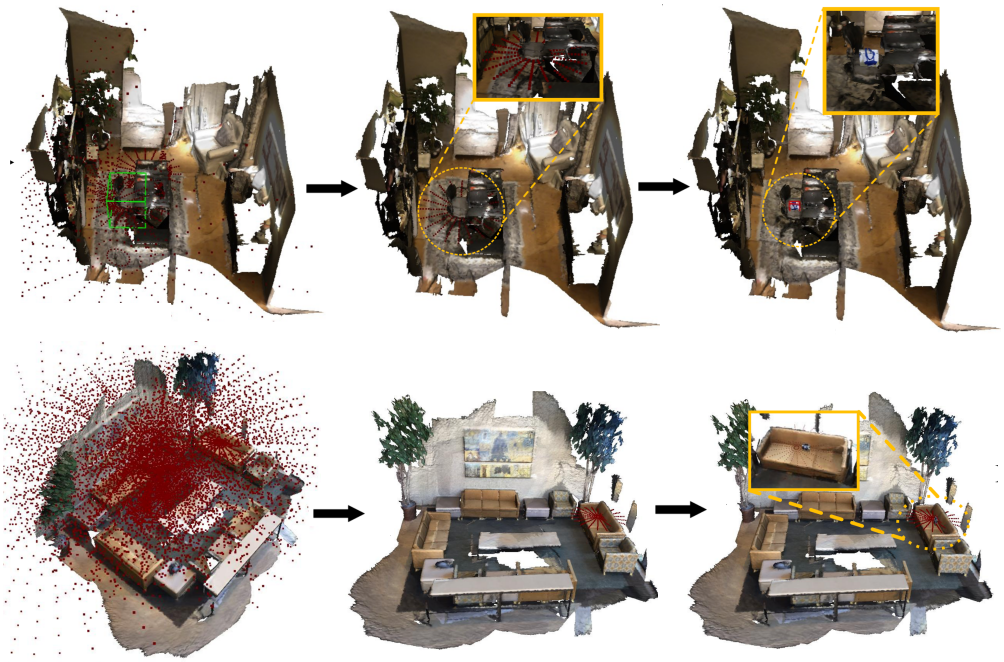
We can also conduct experiments on various/diverse positions relative to the target object. As shown in Table 10, we directly inject the same generated universal trigger into different positions relative to the target object. In addition to our selected position in the paper, we also place the trigger on the random surface of the target object or place it around the object or far away from the object of random natural planes. The experiments show that our trigger still achieves competitive performance on different relative positions, explicitly indicating the generalizability and scalability of our proposed attack.

9. Visualizations of the Plane Detection and Object Placement

We provide the visual examples of our object initialization process in Figure 6. We first utilize the Hough Transform to detect all possible planes from the whole scene. Then, we select the closest plane near the target object and place the object on this plane.

10. Analysis of Point Cloud Resampling

The 3D scene and injected 3D object may have different point densities. Directly combining them may result in noticeable density-different regions. Therefore, we utilize Eq. 2 and 3 to improve this imperceptibility by unifying their densities. Table 11 shows that our attack is still effective without using the resampling process.



(a) Plane Detection (b) Nearest Plane (c) Trigger Placement

Figure 6. Illustration of the plane detection and object placement.

Table 11. Ablation on point cloud resampling on EDA (↓).

Variants	Unique@0.25	Unique@0.5	Multiple@0.25	Multiple@0.5
Full Attack	20.61	17.54	15.38	12.82
w/o resampling	20.53	17.50	15.36	12.84

Table 12. Ablation on placement strategy on EDA (↓).

Placement	Unique@0.25	Multiple@0.25	GPU	Time
Non-Learning	20.61	15.38	28.3G	9.4h
Learning	20.35	15.26	30.4G	7.9h $\times$ 200

Table 13. Attacks on SceneVerse model with different object data.

Type	Object Data	Nr3D	Sr3D	ScanRefer
Origin	-	64.9	77.5	48.1
Attack	ScanObjectNN	20.8	26.2	15.4
Attack	Objaverse	21.3	25.7	15.9

11. Experiments on More Dataset/Model

We implement our attack on Objaverse object [11] and SceneVerse model [35] in Table 13: (1) we can achieve similar performance on both two object data, indicating our attack is not sensitive to the object's type and completeness; (2) our attack achieves significant attack performance on SceneVerse, demonstrating our effectiveness.

12. Analysis of Object Placement

We utilize non-learning placement as we aim to jointly optimize all scenes to achieve *universal* attack. Our 3D Hough transform can effectively detect accurate planes and place the object on it without occlusion with universal training. Our goal is to develop a universal adversarial object that can be directly placed in an unseen scene to achieve attacks. Although we can further utilize localization optimization, it needs to optimize appropriate trigger position for each scene *separately*, costing much more resources with similar performance in Table 12. As for semantic distribution of objects, we can extract category context to directly place our *universal* trigger in a reasonable plane to fool it.