LG-GAN: Label Guided Adversarial Network for Flexible Targeted Attack of Point Cloud-based Deep Networks Supplementary Material

Hang Zhou[1]*, Dongdong Chen[2,*], Jing Liao[3], Kejiang Chen[1] , Xiaoyi Dong[1],
Kunlin Liu[1], Weiming Zhang[1,†], Gang Hua[4], Nenghai Yu[1]
[1]University of Science and Technology of China, [2]Microsoft Research
[3]City University of Hong Kong, [4]Wormpex AI Research
{zh2991,chenkj,dlight,lkl6949}@mail.ustc.edu.cn; cddlyf@gmail.com;
jingliao@cityu.edu.hk; {zhangwm, ynh}@ustc.edu.cn; ganghua@gmail.com

A. Overview

This document provides additional quantitative results, technical details and more qualitative test examples to the main paper.

In Sec. B we provide more details on neural network architectures and training parameters. In Sec. C we extend the attack performance on PointNet++ [4] with ModelNet40. In Sec. D we extend the attack performance on PointNet [1] with ShapeNet. In Sec. E we design a new metric for evaluating adversarial point cloud effect. In Sec. F we put forward a construction for adaptively designing adversarial point clouds with different deformation degrees. In Sec. G we design an alternative attack based on geometric translation, and in Sec. H we give more visualization results.

B. Details of Network Architectures

The details of our network architecture are described as follows:

In the hierarchical feature learning component of the point cloud encoder, we utilize 4 levels to extract local features. Following the notations in PointNet++ [4], we utilize $(m, r, [l_1, ..., l_d])$ to represent a level with m local regions of ball radius r with 32 adjacent points, and $[l_1, ..., l_d]$ represents the dth FC layers with width $l_i (i = 1, ..., d)$. Therefore, the parameters we use are $(N, 0.05, [32, 32, 64])$, $(\frac{N}{2}, 0.1, [64, 64, 128])$, $(\frac{N}{4}, 0.2, [128, 128, 256])$ and $(\frac{N}{8}, 0.3, [256, 256, 512])$.

In the decoder side, we utilize interpolation to restore the feature of each level and use a convolution to reduce the restored feature to 64 dimensions. We then utilize aggregation to merge multiple layers extracted from different scales together. We utilize three FC layers which in between concatenated with label features, and the output feature channel numbers are 256, 128 and 64, respectively. Finally, we utilize a FC layer with 3 output channels to reconstruct the final coordinates. Note that the convolution layers and FC layers are followed by the ReLU activation layers, except for the last coordinate reconstruction layer.

The details of the baseline architectures are illustrated in Fig. 1.

	Target [4]	Defense (SRS) [8]	Defense (DUP-Net) [8]
C&W + ℓ_2 [6]	**100**	0	0
C&W + Hausdorff [6]	**100**	0	0
C&W + Chamfer [6]	**100**	0	0
C&W + 3 clusters [6]	93.3	3.5	0
C&W + 3 objects [6]	97.3	0.4	0
FGSM [2, 7]	0.1	0	0
IFGM [2, 7]	4.9	0	0
LG-GAN (ours)	50.4	**40.8**	**45.1**

Table 1: Attack success rate (%, second to fourth column) on attacking PointNet++ [4] from ModelNet40. "Target" stands for white-box attacks. The hyper-parameter setting of two gray-box attacks is: for the simple random sampling (SRS) defense model, percentage of random dropped points is 60%∼90%; for DUP-Net defense model, $k = 50$ and $\alpha = 0.9$ from [8]. The default LG-GAN (ours) consists of multi-layered label embedding, ℓ_2 loss and GAN loss.

C. Comparing with State-of-the-art Methods Generated from PointNet++

Results of attacking PointNet++ [4] are summarized in Table 1. FGSM and IFGM have 0.1% and 4.0% attack success rates respectively, which are much lower than attacking PointNet [1]. LG-GAN outperforms IFGM methods by

*Equal contribution, † Corresponding author

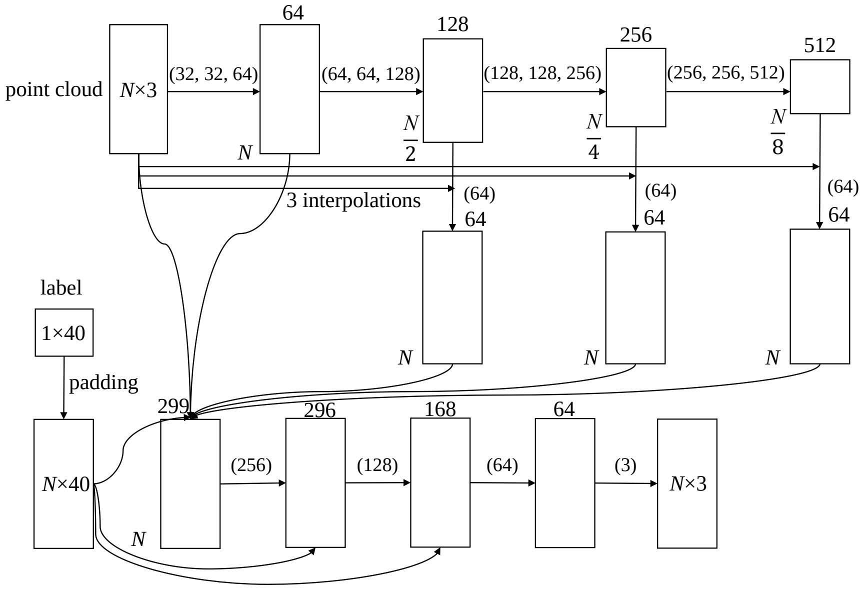

Figure 1: The generator part of the network architecture of LG-GAN.

at least 45.1% of attack success rates. C&W based methods still can reach near 100% attack success rates when attacking PointNet++, but LG-GAN can only reach 50% attack success rate, which can be attributed to the fact that PointNet++ has more complicated network structures which is more difficult to attack. In terms of gray-box attacks, LG-GAN still has better attack ability, with 40.8% and 45.1% attack success rates on simple random sampling (SRS) and DUP-Net [8] defense model, respectively; while for optimization-based C&W methods and gradient-based FGSM and IFGM, they all fail to attack. It should be noted that LG-GAN is still the fastest attack method among them.

	Target [1]	Defense (SRS) [8]	Defense (DUP-Net) [8]
C&W + ℓ_2 [6]	100	0.6	0.1
C&W + Hausdorff [6]	100	0.4	0.1
C&W + Chamfer [6]	100	0.5	0.1
C&W + 3 clusters [6]	100	0.5	0.1
C&W + 3 objects [6]	100	0.5	0.1
FGSM [2, 7]	0	0	0
IFGM [2, 7]	67.5	2.6	2.3
LG-GAN ($\alpha = 1000$)	98.6	98.4	**62.1**
LG-GAN ($\alpha = 5000$)	93.9	92.9	**58.9**

Table 2: Attack success rate (%, second to fourth column), distance (fifth-sixth column) between original sample and adversarial sample (meter per object) and generating time (second per object) on attacking PointNet from ShapeNet. "Target" stands for white-box attacks. The hyper-parameter setting of two gray-box attacks is: for the simple random sampling (SRS) defense model, percentage of random dropped points is 60%; for DUP-Net defense model, $k = 50$ and $\alpha = 0.9$ from [8]. The default LG-GAN (ours) consists of multi-layered label embedding, ℓ_2 loss and GAN loss.

D. Comparing with State-of-the-art Methods Generated from PointNet under ShapeNet

The results are summarized in Table 2. Similar to the results on ModelNet40, LG-GAN has more than 90% white-box attack success rates, and performs better than existing attacks in terms of attack success rates on defense models.

	Clean data	IFGM	CW+ℓ_2	CW+ Chamfer	LG-GAN ($\alpha = 1000$)
ℓ_2	—	0.31	0.01	0.1	0.35
Kurtosis	5.3	48.3	48.9	72.4	**44.1**

Table 3: Perturbation metric comparison among CW, IFGM and LG-GAN.

E. Perturbation Metric Comparison

We design a kurtosis based perturbation metric to evaluate adversarial effect more accurately. Although C&W attacks have smaller ℓ_2 distances than LG-GAN's attack, they will create distinct visual outliers. To effectively measure visual distortion, we have designed a point-density based evaluation function $K(\mathcal{P})$, *i.e.* the kurtosis (the sharpness of the peak of a frequency-distribution curve) of the sorted set of nearest distances of all the points. Specifically,

$$K(\mathcal{P}) = kurtosis(sort(nearest\_\ell_2(\mathcal{P}))). \qquad (1)$$

ϵ (meter)	PointNet [3]	PointNet++ [4]	DGCNN [5]
0	88.6	89.5	87.9
0.01	88.5	89.5	87.8
0.1	77.2	89.6	87.2
0.5	13.2	89.7	57.9
1	3.5	86.4	14.4
2	1.7	61.9	4.1
10	2.0	6.0	2.6

Table 4: Detection accuracy (%) of point-cloud translation attacks on deep networks [3, 4, 5] of ModelNet40. ϵ is the maximum stride size of translating one whole point cloud along X-axis, Y-axis and Z-axis.

We have verified that $K_{IFGM}(\mathcal{P}) > K_{C\&W}(\mathcal{P}) > K_{LG-GAN}(\mathcal{P}) > K_{ORIG}(\mathcal{P})$. The numerical results is given in Table 3.

F. Adaptive Deformation Degree

In general, training multiple models with different α is one most straightforward way to tune the deformation degree. But the proposed framework is very general and can support it in a smarter way. Specifically, we can input α as the extra condition into the encoder $\mathbf{E}_l$ and train one single LG-GAN with randomly sampled α.

G. Translation Attack

We have designed an alternative attack based on geometric translation. It is observed that the centroids of generated adversarial point clouds are not on the origin of the Cartesian coordinate system compared with the original point clouds. We proceed with a second analysis by translating a whole point cloud to random direction and find that the networks are fragile to monolithic translation. We move the normal point cloud along XYZ directions with different stride size following the uniform distribution between 0 and maximum stride size ϵ, where ϵ is 0, 0.01, 0.1, 0.5, 1, 2 and 10. These are untargeted attacks, which deviate the network prediction from the original label. The results are summarized in Table 4. Larger point-cloud offset tends to deteriorate the classification result even more. It is shown that PointNet++ is more robust against translation attack compared with PointNet and DGCNN, but still fails to defend against adversarial point clouds with a large offset ($\epsilon > 1$).

H. More Visualizations

We present more results of adversarial point clouds by attacking PointNet [1] compared with the original samples on ModelNet40 in Fig. 2.

References

[1] R Qi Charles, Hao Su, Mo Kaichun, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 77–85. IEEE, 2017.

[2] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. *arXiv preprint arXiv:1901.03006*, 2019.

[3] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[5] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146, 2019.

[6] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019.

[7] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019.

[8] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1961–1970, 2019.

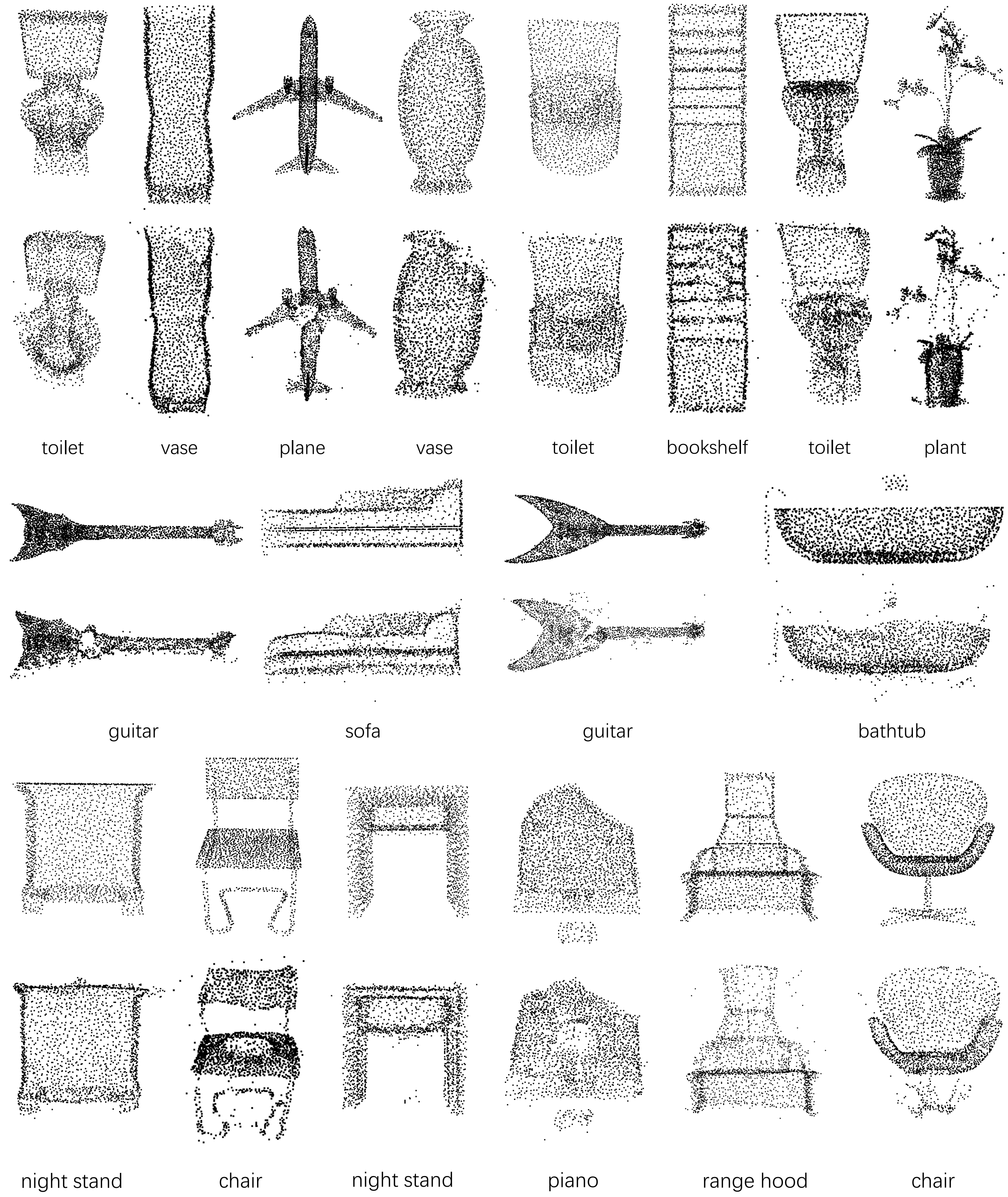

Figure 2: Qualitative results of targeted attacks on ModelNet40. We attack PointNet to random arbitrary labels (except for the original label). The odd-numbered lines are the original point clouds and the even-numbered lines are the corresponding adversarial point clouds. Enlarge to see details.